جامعـــــة حلـــــب
كلية الهندسة الكهربائية والالكترونية
قسم هندسة الحواسيب

# تطوير خوارزميات التنقيب عن المعطيات من أجل مواقع ويب الذكية

# Developing of Data Mining Algorithms for Web Intelligence

رسالة قدمت لنيل درجة الماجستير في هندسة الحواسيب

إعداد
المهندس عصام قزموز

جامعـــــة حلـــــب

كلية الهندسة الكهربائية والالكترونية

قسم هندسة الحواسيب


# تطوير خوارزميات التنقيب عن المعطيات من أجل مواقع ويب الذكية

# Developing of Data Mining Algorithms for Web Intelligence

رسالة قدمت لنيل درجة الماجستير في هندسة الحواسيب


إعـــــداد

**المهندس عصام قزموز**

باشــــــراف

الدكتور يحيى نجار

أستاذ مساعد في قسم هندسة الحواسيب

كلية الهندسة الكهربائية والإلكترونية

الدكتور حيان الحصرم

مدرس في قسم هندسة الحواسيب

كلية الهندسة الكهربائية والإلكترونية

جامعـــــة حلـــــب
كلية الهندسة الكهربائية والالكترونية
قسم هندسة الحواسيب

# تطوير خوارزميات التنقيب عن المعطيات من أجل مواقع ويب الذكية

# Developing of Data Mining Algorithms for Web Intelligence

رسالة قدمت لنيل درجة الماجستير في هندسة الحواسيب

أعدها الطالب
**المهندس عصام قزموز**

بإشراف
**الدكتور المهندس يحيى نجار**
**الدكتور المهندس حيان الحصرم**

تاريخ مناقشة الرسالة ١٩ /٢/٢٠٠٩م

قدمت هذه الرسالة استكمالاً لمتطلبات نيل درجة الماجستير في هندسة الحواسيب من كلية الهندسة الكهربائية والإلكترونية بجامعة حلب.

Submitted in partial fulfillment of the requirements Master Degree of Computer Engineering at the Faculty of Electricity and Electronic Engineering, Aleppo University

**الجمهورية العربية السورية**

**جامعة حلب**

## شهـــــادة

نشهد بأن العمل الموصوف في هذه الرسالة هو نتيجة بحث علمي قام به المرشح المهندس عـــصام قزمـــوز تحت إشراف الدكتور المهندس يحيى نجار في قسم هندســـة الحواســـيب مـــن كليـــة الهندســـة الكهربائيـــة والإلكترونية بجامعة حلب والدكتور المهندس حيان الحصرم في قسم هندسة الحواسيب من كلية الهندسة الكهربائية والإلكترونية بجامعة حلب

وأي رجوع إلى بحث آخر في هذا الموضوع موثق في النص.

| المرشح | المشرف | المشرف |
|---|---|---|
| عصام قزموز | يحيى نجار | حيان الحصرم |

## CERTIFICATE

It is hereby certified that the work described in this thesis is the result of the author's own investigation under the supervision of Dr. Y. Najjar in the Department of Computer Engineering, Faculty of Electricity and Electronic Engineering, Aleppo University and Dr. H. Housroum in the Department of Computer Engineering, Faculty of Electricity and Electronic Engineering, Aleppo University and any references to other research work has been duty acknowledged in the text.

Directors of study
Dr. Y. Najjar
Dr. H. Housroum

Candidate
Eng. I. Kazmouz

## تصريـــــح

أصرح بأن هذا البحث " تطوير خوارزميات التنقيب عن المعطيات من أجل مواقع ويب الذكية" لم يسبق أن قبل لأي شهادة ولا هو مقدم حالياً للحصول على شهادة أخرى.

المرشح

عصام قزموز

## DECLARATION

It is hererby declared that this work has not already been accepted for any degree, and it is not being submitted concurrently for any other degree.

Cardidate

I. Kazmouz

## الفهـــرس

# الفصل الأول

## المقدمة introduction

**١-١تمهيد:**

مع التطور العلمي الكبير في كافة الميادين ، و خاصة في مجال الكتب الإلكترونية المنتشرة عبر مواقع الويب ، كان لا بد من البحث عن طرق فعالة لإيجاد خوارزميات بحث قوية و طرق تخزين للمعلومات في قواعد البيانات للحصول على أفضل النتائج من الموضوعات المفيدة الخاصة و المرتبطة بالعنوان المراد البحث عنه قدر المستطاع دون تشتت.و بالتالي الحصول على المعلومة بكتابة كلمات مفتاحيه معينة و ذلك باستخدام تقنيات جديدة ومفيدة.

وإن الطرق المستخدمة في بناء مواقع الويب وبناء قواعد المعطيات و طرق التخزين كثيرة ومتنوعة و عمليات البحث و الخوارزميات المستخدمة أيضاً متنوعة ومنها محركات البحث Yahoo & Google إلا دليلاً على التطور الهائل في عمليات البحث للحصول على معلومات فعالة بجهد قليل ، و يتعلق هذا الأمر بطريقة بناء قاعدة المعطيات و المعلومات الأساسية المطلوب تخزينها عن المواقع الموجودة و الخوارزميات المستخدمة في البحت والطرق المستخدمة في التنقيب عن المعطيات .

ولقد نشأت فروع جديدة في مجالات الذكاء الصنعي للقيام بذلك وكمثال عليها التعلم الآلي،والتنقيب عن المعطيات Machine Learning و Data Mining [٨-٥] التي تركز على استنباط المعرفة من المعطيات الضخمة .

ولهذه الفروع علاقة وثيقة بفروع أخرى من العلم منها علم الإحصاء وقواعد المعطيات ، ولكن يوجد فروق بين هذه العلوم وبين فروع الذكاء الصنعي ، حيث إن علم الإحصاء يركز على اختبار الفرضيات أما Machine Learning فيركز على تحسين الأداء لتعليم الأداء ، وكذلك الأمر له علاقة بالتعليم في الزمن الحقيقي وتعليم الروبوتات .

بينما يقوم علم التنقيب عن المعطيات وتغطية المعرفة في قواعد المعطيات Data Mining And Knowledge Discovery In Database بالتركيز على استنباط المعرفة من قواعد المعطيات الصحيحة ولهذا الفرع عدة مهام :

١- التصنيف Classification [١-٤]:مثل عملية تحديد الأصناف بشكل مباشر.

٢- التجميع Clustering [٢٠]:تصنيف الأمثلة إلى عدد من المجموعات بشكل أوتوماتيكي.

٣- الارتباط Associations [٢٤-٢٥]:اكتشاف النماذج في أنظمة التعلم بدون إشراف.

**١-٢ مجال اهتمام هذا البحث The Project Scope:**

المجال المحوري لاهتمام هذا البحث هو التنقيب في المحتوى النصي للويب، أي مستندات الويب النصية.

حيث يعتمد التنقيب النصي على استخدام معالجة اللغات الطبيعية والتقنيات الإحصائية / الاحتمالية/ لإيجاد الكتل والنماذج المعرفية في التجمعات النصية.

**الدوافع أو المبررات (motivation):**

١. ضخامة الويب و التنوع الهائل للمحتويات.

٢. التوزيع المتشتت وصعوبة الوصول إلى المعلومات المطلوبة بدقة.

٣. فقدان الهيكلية، وعدم التجانس في الأوساط المستخدمة.

٤. التغير الديناميكي والمستمر للمحتويات.

٥. الانفجار الهائل للمعلومات.

**١-٣ أهمية البحث وأهدافه (Aims and Objectives):**

يعرض هذا البحث طرق التصنيف التي تعتمد على خوارزميات بناء الأشجار وأهمية هذه الطرق في عملية التصنيف مع التركيز على طريقة التصنيف لخوارزمية C٤,٥ التي تعد من خوارزميات التعلم

بإشراف وتكمن أهميتها في دقة تصنيفها وتغطيتها لكافة القوانين، مع توضيح الزمن المستهلك من قبلها لحساب الربح والأنتروبي عند كل عقدة مقارنة.

وبالتالي تم طرح تعديل على هذه الخوارزمية من أجل تقليل عدد حسابات الأنتروبي وبالتالي تقليل الزمن اللازم للحساب وذلك من خلال حساب الربح لبعض الحالات وفق شرط مقترح وعدم حساب الأنتروبي والربح لكل العقد وبالتالي قد نصل لعقد لا نحتاج لحساب الربح عندها.

كما وقد تم طرح تعديل آخر على خوارزمية C٤,٥ من أجل تشذيب القوانين الناتجة عنها للوصول إلى قوانين أكثر شمولية وتشذيباً (Pruning) [٢٣-٢١].

تم إعداد برنامج بلغة C# للتعديلين المقترحين.

**١-٤ الخطوط العامة للأطروحة (Outline of the Thesis):**

إن هذه الأطروحة تتألف من ستة فصول. يضم الفصل الأول مقدمة عن البحث وأهدافه وأهميته.سيتم التحدث في الفصل الثاني عن مقدمة عامة عن الويب وأساسيات البحث ضمنه ومهام التنقيب عن المعطيات في الويب.

سيتم التعريج في الفصل الثالث عن البنية التحتية للويب والتكلم عن الزاحف Crawler.

سيتم التطرق في الفصل الرابع إلى التنقيب في المعطيات مهامها ووظائفها ومن ثم التعرف إلى سيناريو التعلم بإشراف.في الفصل الخامس سيتم عرض أشهر خوارزميات التصنيف المستخدمة.

سيتم في الفصل السادس دراسة التعديلات المقترحة وتحليل النتائج التي تم التوصل إليها في البحث ومقارنتها مع الطرق الأخرى المستخدمة. وأخيراً سيتم في الفصل السابع عرض ملخص عن البحث وعن الأعمال المستقبلية لتطوير البحث.

# الفصل الثاني

## مقدمة عامة عن الويب و التنقيب عن المعطيات

١. أساسيات البحث في الويب وطرق الوصول إلى المعلومة في الويب.

٢. تحديات تصميم ويب ذكي.

٣. مهام التنقيب عن المعطيات.

### ٢- ١الطرق المستخدمة للوصول إلى المعلومة المخزنة في الويب:

يُستخدم علم التنقيب عن المعطيات في فروع عديدة لعل أهمها Web Mining[١١،٢٦]، حيث يقوم الناس حالياً باستخدام طرق أساسية للوصول إلى المعلومات المخزنة في الويب و هذه الطرق هي :

- البحث المعتمد على كلمات مفتاحيه Keyword-Based Search Or Topic-Directory Browsing باستخدام محركات بحث مثل Google & Yahoo ،إذ نقوم باستخدام الكلمات المؤشرة أو الفهارس المبنية يدوياً لإيجاد المقالات أو المواضيع وفق الكلمات المفتاحية أو الموضوعات المحددة المراد البحث عنها.
- القيام بالاستعلام عن المعلومات من مصادر الويب العميقة و الداخلية مثل المواقع التي تحتوي على الكتب و كمثال عليها amazon.com's & realtor.com's .
- المتصفح العشوائي Random Surfing التي تتبع في البحث و تعتمد على مؤشرات روابط الويب .

### ٢- ٢التحديات التي تواجه تصميم ويب ذكي:

يبقى هناك الكثير من التحديات لإيجاد مثل هكذا تصميم للويب الذكي ومن هذه التحديات[١١،٢٦] :

- المستوى التجريدي Abstraction Level ، المخطط التقليدي للوصول لكميات هائلة من المعطيات القيمة الموجودة بشكل أساسي داخل الويب، مفترضةً عرض أغراض النص المعتمد على الكلمات المفتاحية القاعدية لصفحات الويب.
- مستوى التخديم Service Level، حيث علينا استبدال مخطط الوصول الأمثلي بنسخ معقدة أكثر و التي يمكن أن تستغل الويب بشكل كامل .

### ٢-٣مهام التنقيب عن المعطيات في الويب Data Mining Tasks in Web:

طالما سنركز في دراستنا على موضوع التنقيب عن المعطيات من أجل مواقع الويب الذكية Data Mining For Web Intelligence فلا بد من شرح عن بعض مهام التنقيب عن المعطيات في الويب :

١) التنقيب عن المعطيات في محركات البحث Mining Web Search-Engin Data:

إن محركات البحث في الويب تعتمد على الفهرسة ( صفحات الويب المنتشرة ) و بناء مخزون المعطيات المعتمدة على مؤشرات الكلمات المفتاحية التي تساعد على تحديد مجموعة من صفحات الويب التي تتضمن كلمات محددة باستخدام مجموعة من الكلمات أو العبارات المعقدة.

٢) تحليل بنية روابط الويب Analyzing The Web's Link Structure:

إن بنية روابط الويب تملك عدة ميزات و هذه الميزات هي :

أ- لا تقوم كل روابط الويب بعملية دعم ومصادقة البحث ، حيث يتم عادةً إنشاء بعض الروابط لأغراض أخرى مثل عمليات التحري أو الإعلانات المدفوعة.

ب- المستندات التي تتجنب الاهتمام المنافس و التجاري سيكون نادراً امتلاكه مؤشر صفحة الويب إلى الصفحات مرجعية، كمثال إن موقع Coca-Cola سيكون من المحتمل تجنبها لرابط صفحة Pepsi .

ت- قلما تؤمن صفحات المستندات التوثيقية وصفاً واضحاً.

٣) تصنيف صفحات الويب أوتوماتيكياً Classifying Web Documents Automatically :

إن الوصول إلى إنقاص التكلفة و زيادة السرعة للتصنيف الأوتوماتيكي العالي يشكل الشغل الشاغل لمصممي صفحات الويب حيث تعتمد طرائق التصنيف النموذجية على استخدام أمثلة ايجابية وأخرى سلبية كمجموعات تدريب ثم تقوم بتحديد كل عنوان مصنف إلى مجموعته التابع لها.

٤) التنقيب داخل صفحة المعطيات و محتويات الصفحة Mining Web Page Semantic Structures And Page Contents :

إن صفحات الويب تختلف فيما بينها حسب ما يلي :

أ- تعريف نتيجة جيدة ومناسبة للحصول على الخلاصة. وذلك إما بتحديد هذه البنيـــة الخبيـــرة بشكل يدوي من أجل صف ويب معطى ، أو تطوير تقنيات لإحداث بنية مـــن مجموعـــات صفحات ويب بشكل أوتوماتيكي.

ب- إن المطورين يمكن أن يستخدموا بنية صفحة ويب وطرائق تصفية للمحتويـــات مـــن أجـــل الوصول الأوتوماتيكي إلى صفحة الويب.

ت- بنية صفحة الويب والتعرف على محتوياتها ستعزز بشكل كبير عمـــق تحليـــل محتويـــات الصفحة وبناء طبقات متعددة من معلومات قاعدة الصفحة.

٥) صفحات الويب الديناميكية Mining Web Dynamics :

إن التنقيب في الويب يمكن أن يحدد صفحات ويب ديناميكية وكيفية تغيرات محتويات بنية صفحات الويب وكيفية نماذج الوصول بالإضافة إلى تخزين قطع معينة من المعلومات المتعلقة ببرامترات التنقيب عن الويب وذلك للمساعدةً في اكتشاف التغيرات في محتويات الويب وارتباطاته.

٦) بناء صفحات متعددة الطبقات و الأبعادBuilding Multilayered ,Multidimentional Web:

يمكن أن نستخدم صفحات متعددة الأبعاد في عدة خطوات رئيسية:

أ- يمكن أن تحلل مجموعة من صفحات الويب وهذا التحليل ســيغطي محتويـــات الـــصفحات وبنيتهـــا، والارتباطات و نماذج الاستخدام وهذا بدوره سيمّكن من :

أ-١- تجميع مجموعة متكاملة من صفحات ويب محلية مرتبطة في مقاطع تدعى Semantic Page.

أ-٢- التعامل مع صفحات مستقلة إذا كانت النماذج لها مقاطع مستقلة.

ب- يمكن أن نركب معلومات ويب متعددة الطبقات والأبعاد بشكل مباشر اعتماداً على نظم خبيـــرة مـــزودة بقواعد معطيات .

وبالتالي يمكننا استخدام هذا النظام المبني للاستفسار عن المعلومات و تحليل التنقيب عن المعطيات.

إن إنشاء صفحات ويب تحتوي نسخاً عن كل صفحة على الإنترنت سيكون أمراً شاقاً لأنــه يتطلــب إنشاء نسخة على كامل الويب.

وبالتالي يمكن وكمثال بسيط أن نقوم بتعريف مستند كما يلي:

Document(file_addr,doc_category,authoritative_rank,key_words,authors,title,journal_or_book_title,publication,abstract,language,table_of_contents,categorydescription,index,links_out,multimediaattached,num_pages,form,size_doc,line_tamp,…..,access_frequency)

وبهذا التعريف كل مدخل يمثل تجريداً لصفحة الوثيقة، مثلاً إن أول خاصية file_addr يقوم بتخزين اسم عنوان URL الخاصة بالانترنت.......

٧) مهام أخرى للويب الذكيOther Data Mining Tasks For Web Intelligence:

هناك طرائق تنقيب عن المعطيات واعدة تستطيع المساعدة في الحصول على صفحات ويب ذكيــة وفعّالة والقيام بخدمة الزبائن من أجل ملاحقة طلبات خاصة ومستقلة .

ولكن حتى هذا التاريخ بعض صفحات الويب بنيت اعتماداً على نظم الخدمة التجارية و على الرغم من أن بناء هذه الصفحات التجارية الشخصية يساعد على تقديم خدمة مناسبة ، فإن هذا النظام لن يجمع معلومات كافية حول التوصية بمعلومات نوعية و مستقلة.وبالتالي إن طريقة البحث الآنفة الذكر ستنتج مشكلتين وهما:

١- بعض نتائج البحث لا علاقة له بطلب المستخدم.

٢- العديد من المقالات المتماثلة ستكون نتيجة البحث.

.

# الفصل الثالث

١. البنية التحتية للتنقيب في الويب.

**Web Mining Infrastructure.**

٢. ما هو التنقيب في الويب؟

**What is Web Mining?**

٣. أنواع التنقيب في الويب.

**Web Mining Taxonomy.**

٤. أساسيات البحث على الويب.

**Web Search Basics.**

٥. الزحف عبر الويب.

**Web Crawling.**

## ٣-١البنية التحتية للتنقيب في الويب Web Mining Infrastructure:

**تمهيد:**

تُعتبر شبكة الويب العالمية (World Wide Web) أكبر وأضخم مستودع لمختلف أنــواع المعطيــات والمعلومات الحاسوبية، ولا سيّما النصوص التشعبية (hypertexts).

اختصاراً الويب أو www، تضم هذه الشبكة مليارات صفحات الويب التي نُشرت و تنشر من قبــل ملايين الناس، تُحرر وتُعدل من قبل أكثر من جهة، وتُوزع على ملايــين الأجهزة ومخدمات المواقع.

أدت هذه الآلية غير المضبوطة في العمل، بالإضافة إلى التضخم المستمر إلى عــدم القــدرة علــى احتواء الويب بأي شكل من الأشكال، لا أعتقد أن هناك أحداً لا يعرف الإنترنت أو الويب، أو لــم يــسمع بضخامة المشكلة وتنوعها. ومما يثير الدهشة والاستغراب أن تلك التوليفة العجيبة التي اسمها الويب تعمل!!.

سأقدم في هذا الفصل شرحاً عن موضوع البحث، على أن تتم مناقشة التفاصيل في الفصول اللاحقة.

يضم هذا القسم الأجزاء التالية:

١. ما هو التنقيب في الويب؟ What is Web Mining?

٢. أنواع التنقيب في الويب. Web Mining Taxonom

٣. أساسيات البحث على الويب. Web Search Basics

٤. الزحف عبر الويب. Web Crawling

### ٣- ٢ ما هو التنقيب في الويــب:

**What is Web Mining?**

"إن كان البستانيُّ هاوياً أو محترفاً، فإنه يعلم تماماً كيف يسلك دربه في حديقته".ولكن للأسـف هـذه المقولة ليست قابلةً للتطبيق على الويب إطلاقاً، لسبب بسيط وهو أن غالبيـة الـصفحات والمواقـع يمكـن الوصول إليها مثلها مثل البقية (بنفس درجة الاحتمالية تقريباً). وما تزال خدمات البحث التقليدية إلى الآن لا تفترض بشكل مباشر وجود خلفية محددة أو دراية بكيفية البحث لدى المستخدمين.

إذاً نحن بحاجة إلى سلوك الطريق الصحيحة خلال تجوالنا في الويب. بالتالي يجب علينا اكتشاف هذه الطريق أولاً، ومن ثم محاولة البحث في المعلومات التي تهمنا.

وهذا ما توفره لنا تقنيات التنقيب في الويب.

#### ٣-٢-١ تعريف التنقيب في الويب (Web mining):

اشتقاق نماذج المعرفة المخفية /والتي من المحتمل أن تكون مفيدة/ من مختلف محتويـات و أنـشطة شبكة الويب العالمية.

أو استخدام التقنيات الخاصة بمجال التنقيب في المعطيات،لاكتـشاف و اسـتخلاص المعرفـة مـن المستندات والخدمات المتوفرة على الويب، وذلك بشكل أوتوماتيكي.

#### ٣-٢-٢ ما الفرق بين التنقيب في المعطيات والتنقيب في الويب؟

الفرق الأساسي يكمن في أن التنقيب في المعطيات هدفه استخلاص المعرفة من المعطيـات المهيكلـة المخزنة في قواعد المعطيات.

في حين أن التنقيب في الويب يهدف إلى استخلاص المعرفة من بيانات أقل هيكلة (شـبه مهيكلـة) أو غير مهيكلة إطلاقاً.

بالإضافة إلى وجود ثلاثة نطاقات للتنقيب في الويب سيأتي ذكرها في الفقرة التالية.

من الفروق الأساسية بين التنقيب في المعطيات والويب أيضاً، استخدام تقنيات استرجاع المعلومات (information retrieval techniques) وخاصة النصية في التنقيب في الويب بمعنى أنها أصبحت ( Web information retrieval techniques)

و من أكثر الأمثلة وضوحاً على النظم التي تستخدم هذه التقنيات محركات البحث بمختلف أنواعها AltaVista, Google, Yahoo!.

### ٣-٣ أنواع التنقيب في الويب و تصنيفاته Web mining taxonomy:

تصنف Web Mining إلى ثلاث طرائق أساسية :

١-التنقيب في محتويات الويب Web content Mining.

٢- التنقيب في بنية الويب Web structure Mining.

٣- التنقيب في حالات استخدام الويب Web usage Mining.

والدراسات الحديثة تحاول إيجاد تضافر تقني بين طرائق نظم المعرفة والويب الذكي Knowledge-based system and Web-mining.

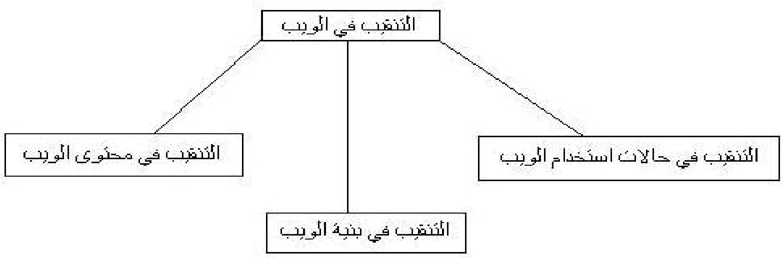
حيث إن Knowledge-based systems تتضمن نموذجين :

١- حيث يُبنى من القوانين والحقائقKnowledge module .

٢- نموذج التحكمControl module .

سيتم استخدام تقنية web-mining لاستنتاج القوانين والحقائق من صفحات الويب.

كما يبين الشكل التالي أنواع التنقيب في الويب.




الشكل رقم (٣-١)

**أ- التنقيب في محتوى الويب Web content mining:**

التنقيب في محتوى الويب هو عملية أوتوماتيكية لا تقتصر على استنساخ الكلمات المفتاحية فقط، وبما أن محتوى المستندات النصية لا يمثل لأية بنية واضحة ومقروءة بالنسبة للحواسب، فإنه يجب اعتماد طرق وأساليب للتعامل مع هذا المحتوى بحيث تستطيع الحواسب استيعابه. وهذا يفيد بالدرجة الأولى في مـــساعدة المستخدمين على إيجاد المستندات التي يبحثون عنها.

هناك مجموعتان أساسيتان من استراتيجيات التنقيب في المحتوى:

١. استراتيجيات تتعلق باسترجاع المعلومات Information retrieval strategies:

والتي تعتمد على مايلي:

- تمثيل المعطيات بشكل غير مهيكل (نص حر).
- تحليل النص ودراسته. (text analysis).
- تصنيف صفحات الويب، وتجميعها. (pages classification, clustering).

٢. استراتيجيات تتعلق بمجال قواعد المعطيات:

- تمثيل المعطيات بشكل شبه مهيكل (semi structured).
- استخدام نموذج تبادل الأغراض.
- إيجاد مخططات ويب، بُنى ويب، دعم استفسارات من نمط SQL.

**ب- التنقيب في البنية الهيكلية للويب Web structure mining:**

يمكن استخراج معلومات أكثر من مجرد المعلومات المحتواة في صفحات الويب كمحتوى نـــصي أو وسائط متعددة. على سبيل المثال، الارتباطات التشعبية (hyperlinks) التي تشير إلى صفحة معينة (in links) تمثل دليلاً على مدى أهمية هذه الصفحة، بمعنى أنه كلما زاد عدد الارتباطات التشعبية الداخلة أو المؤدية إلى تلك الصفحة زادت أهمية الصفحة.

وبالمثل، فإن عدد الارتباطات الخارجة من الصفحة قد يمثل دليلاً على مدى غنى مواضيع الـصفحة، أي أنها تغطي مواضيع كثيرة.

يستخدم Google خوارزمية لتقييم مدى جودة الصفحة ونوعيتها تسمى ترتيـب الـصفحة Page Rank (Page Rank and HITS).

تفترض هذه الخوارزمية ما يلي:

تكون الصفحة مهمة إذا ما ارتبطت بها مجموعة من الصفحات المهمة.

ارتباط b → a:

- a تعتبر b مهمة.
- كلما زادت أهمية a زادت أهمية b.

يوفر الملحق A شرحاً مبسطاً عن آلية عمل الخوارزمية.

**ت- التنقيب في حالات استخدام الويب Web usage mining:**

تقوم مخدمات الويب بتسجيل مراكمة المعطيات حول التنقلات والتعاملات التي يقوم بها المستخدمون حالما يتم استقبال طلبات تخديم من أولئك المستخدمين.

يساعد تحليل سجلات الوصول إلى الويب لمختلف المواقع على فهم أفضل لسلوك المستخدمين وبنيـة الويب.

هناك اتجاهان للتنقيب في حالات الاستخدام:

١. تعقب نماذج الوصول العامة General access pattern tracking.
٢. تعقب حالات الاستخدام المخصص Custom usage tracking.

## ٣-٤ أساسيات البحث على الويب Web Search Basics:

شبكة الويب هي ظاهرة فريدة وغير مسبوقة من عدة نواحٍ:

١- غير مسبوقة من ناحية الحجم.

٢- والافتقار الكامل للتنسيق أثناء تضخمها،واختلاف الخلفيات والدوافع للمشاركين في زيادة حجمها، وغيرها الكثير.

تساهم جميع العوامل المذكورة في جعل عملية البحث على الويب مختلفة وبشكل عام أصعب كثيراً من عمليات البحث التقليدية في المستندات.

**٣ - ٤-١ خصائص الويب Web characteristics:**

من أكثر صفات الويب وضوحاً هي النمو الانفجاري للمحتوى وعمليات نشر المحتوى بشكل غير مركزي،هذا بالإضافة إلى غياب الإدارة المركزية للتحكم في تأليف وتحرير المستندات (أي شخص يستطيع أن يمتلك و يدير موقعه بنفسه ويصبح محرراً لمحتوياته دون رقابة أو إدارة مركزية).

وهذا ما اتضح جلياً من خلال السعي الحثيث لمحركات البحث في عمليات الفهرسة و الاسترجاع، لتذليل العقبات السابقة.

أدى اعتبار الويب ساحةً "مفتوحة" إلى تغيير مفهوم النشر والتحرير بشكل جذري. فأصبح بإمكان الأفراد تأليف وتحرير ونشر إنجازاتهم على الويب بشكل صريح، والأفضل من ذلك أنه يتم بشكل سهل جداً، بعدما كان هذا النطاق من الأعمال الأدبية والكتابية والعلمية حكراً على الأشخاص المؤهلين والمدربين للقيام بذلك.

طبعاً رافق ذلك التنوعُ الهائل في الأساليب الدِلالية للنصوص ومدى جودة صياغة هذه النصوص. ناهيك عن الاختلاف في الأساليب النحوية واللغوية وغيرها التي سببت أقل ما يمكن القول عنه كابوساً لخبراء اللغات والمعلوماتية واسترجاع المعلومات.

ولكن كما يقال "الحاجة أم الاختراع". تضافرت جهود الخبراء في مختلف المجالات لابتكار الطرق والأساليب للتعامل مع تلك القضايا المستجدة.

بينما يبقى السؤال "ما هو حجم الويب؟"الإجابة عن هذا السؤال صعبة المنال، فإنه من الأدق أن نسأل "ما عدد صفحات الويب التي يفهرسها (الموجودة في فهرس) محرك بحث؟"

في نهاية العام ١٩٩٥، صرَّح خبراء محرك البحث AltaVista أنه قد تم فهرسة ما يقارب من ٣٠ مليون صفحة ويب ثابتة (ستاتيكية)، حيث لا يتغير محتوى هذه الصفحات بشكل متكرر.

بناءً على ذلك، يمكن اعتبار الموقع الخاص بمحتوى تعليمي (قليلاً ما يتغير)، موقعاً ثابتاً. بينما الموقع الذي يسجل حركة مطار (شركة طيران) لا يمكن اعتباره ثابتاً البتة، وإنما يدعى موقعاً ديناميكياً (من ناحية المحتوى، وليس من الناحية التفاعلية مع المستخدم، أي أن محتواه يتغير باستمرار).

**٣ – ٤ –٢ التلاعب بمحتوى الويب Spamming:**

ما دفعني إلى الحديث عن ظاهرة الـ spam في عالم الويب، هو طرافة الأساليب و خبثها في نفس الوقت التي تستخدم في عمليات التحايل.

أدى نجاح تقنيات ومحركات البحث على الويب إلى استقطاب انتباه المنحى الاقتصادي إلى تلك الوجهة الجديدة، ولنقل الثورة الجديدة في عالم المعلوماتية والبحث فأصبح الاقتصاديون يرون في محرك البحث وسيلةً لإيصال الإعلانات (تجارية و غير تجارية) إلى المشترين والزبائن المتوقعين.

ما زالت محركات البحث تعتمد تقنية تثقيل المصطلحات ومعدلات تكرارها في صفحات و مستندات الويب لإيجاد مرتبةٍ لصفحة ما تدل هذه المرتبة على أهمية هذه الصفحة.

ساعد هذا على ظهور ما يسمى الـ spam، وهي تُعرَّف (من خلال سياق البحث على الويب):

**٣–٤–٣ تعريف التلاعب بمحتوى الويب spam:**

التلاعب بمحتويات مستند ويب لجعل محركات البحث ترفع من مرتبة المستند في عمليات البحث من أجل كلمات مفتاحية منتقاة.

نستطيع القول إن هناك حرباً دائرة بين محركات البحث والمسؤولين عن عمليات التلاعب تلك، لكشف الأساليب المتبعة لخداع محركات البحث.

يقوم المتلاعبون (spammers) بزيادة تكرار بعض الكلمات التي تعتبر مفتاحية في النص والتـي قـد يعتمد عليها محرك البحث في تصنيف وترتيب الصفحات.

طبعاً هذا لا يقتصر فقط على كلمات البحث المتعلقة بنص المستند، وإنما قد يقوم متلاعب مـا بـذكر مصطلحات لا تتعلق بمحتوى الصفحة أساساً، وبالتالي سيقوم محرك البحث برؤية تلك المصطلحات ومن ثم فهرستها ورفع مرتبة الصفحة.

ولكن استطاعت محركات البحث أن تكتشف هذا الأسلوب. فلجأ المتلاعبون إلى جعل لون تلك الكلمات المفتاحية (الفائضة) بلون خلفية الصفحة، بحيث لا يستطيع المستخدم رؤيتها و بنفس الوقت بإمكان محـرك البحث رؤيتها وفهرستها.

وكما في نظام الصفحات الصفراء في دليل الهاتف، تكون الخدمة مدفوعة التكاليف حيث إنه كلما كبر حجم الخط وأخذ مساحةً أكبر، زادت التكاليف. وهذه حال بعض خدمات محركات البحث الدعائيـة التـي لا يمكن لمستخدم محرك البحث (الباحث) أن يتصورها. فعلى سبيل المثال، هناك ما يسمى التضمين المـدفوع (paid inclusion)، بحيث تدفع تكاليف فهرسة صفحات موقع لمحرك بحث.

تختلف محركات البحث في استراتيجياتها في الكثير من الأمور، ومنها في مسألة التضمين المـدفوع، هناك بعض محركات البحث تمنع التضمين المدفوع اعتقاداً منها بأن ذلك سيؤثر في عمليات البحث ونتائجه.

توجد تقنية أكثر تعقيداً يتبعها المتلاعبون،وهي الحجب (cloaking)، تقتضي بأن يعيد مخـدم الويـب الخاص بالمتلاعب صفحات مختلفة اعتماداً على ما إذا ورد طلب HTTP من مفهرس محرك ويب، أو مـن مستعرض عادي كمستعرض Internet Explorer. وعندما يبحث المستخدم عن كلمات مفتاحية محددة ويحاول استعراض الصفحات التي تحتوي المصطلحات، فإن الصفحة المعادة تختلف تماماً عن تلك المفهرسة.

هنالك أيضاً ما يسمى صفحة مدخل (doorway page)، تحتوي على نص و معلومات مبدئيـة ( meta-data) منتقاة بشكل حذر حتى يتم رفع ترتيب الصفحة في محرك البحث.

عندما يطلب مستعرض ويب صفحة مدخل، يعاد توجيهه إلى صفحة أخـــرى ذات طبيعـــة دعائيـــة و مختلفة المحتوى.

أعتقد أننا في هذه اللحظات قد أعطينا فكرة لا بأس بها عن مدى الصعوبات التي نواجهها في عمليات البحث و التنقيب في الويب.

ساهمت الأساليب السابقة ،باعتبارها ذات دوافع اقتصادية، في ظهور صناعة تسمى محسِّنو محركات البحث (search engine optimizers)، هدفهم تقديم النصيحة للزبائن الذين يريدون رفع مرتبة صفحاتهم ضمن فهارس محركات البحث. و هذا ما أثار حمية المسؤولين عن محركات البحث، و اعتبارهم تلـــك الخـــدمات خرقاً لحقوقهم.

**٣ –٤– ٤ خبرة المستخدم الذي يقوم بالبحث The search user experience:**

لمع اسمع محرك البحث Google في مجال البحث بشكل كبير، و المنافسة مازالت محتدمة بينه و بين بقية محركات البحث.

و لكن كيف استطاع Google أن يتخذ خطوات سباقة على منافسيه؟

حدد Google معيارين أساسيين اعتمدهما:

١. التركيز على مدى ارتباط نتائج البحث بكلمات البحث المفتاحية، وبخاصة الدقـــة فـــي النتـــائج الأولى.

٢. الاهتمام بمدى خبرة المستخدم الذي يقوم بعملية البحث، والافتراض بأنها فقيرة (علماً أنه يـــوفر خدمات متقدمة ومحددة في عمليات البحث).

الفائدة من المعيار الأول:

توفير الوقت على المستخدمين في تحديد المعلومات التي يبحثون عنها، بعرضها في المقدمة.

الفائدة من المعيار الثاني:

تزويد المستخدم بخبرة مرنة.

**٣ – ٤ –٥ حاجات المستخدم الاستفسارية User query needs:**

هناك ثلاث فئات أساسية يمكن أن نصنف استفسارات البحث ضمنها:

١. استفسارات معلوماتية Informational queries.

٢. استفسارات ملاحية (متعلقة بالتجوال) Navigational queries.

٣. استفسارات تتعلق بالصفقات (المناقلات) Transactional query.

**<u>الاستفسارات المعلوماتية:</u>**

استفسارات هدفها البحث عن معلومات عامة في موضوع واسع، مثل البحث عن مرض اللوكيميا.

**<u>الاستفسارات الملاحية:</u>**

استفسارات هدفها البحث عن موقع محدد أو الصفحة الرئيسية لشيء (كائن) منفرد يعلم المستخدم ماهيته (لا يحتوي الويب فقط على مواقع!!)، مثل البحث عن موقع الخطوط الجوية السورية.

**<u>الاستفسارات المتعلقة بالصفقات:</u>**

استفسارات تُعتبر مقدمة للصفقات التي يقوم بها المستخدمون على الويب، مثل شراء منتج، تحميل ملف.

وعملية تحديد فئة الاستفسار ليست سهلة على الإطلاق، وذلك لأنها لا تحكم فقط نتائج البحث الخوارزمية، وإنما مدى قبول الاستفسار وخصوصاً إذا كان يتضمن نية للشراء أو البيع.

**٣–٥ الزحف في الويب Web Crawling:**

**<u>تعريف الزحف في الويب (Web crawling):</u>**

العملية التي يقوم فيها النظام بجمع ما أمكن من صفحات (مستندات) الويب المفيدة، بغرض فهرستها و دعم محركات البحث.

الغرض من عملية الزحف هو تجميع أكبر عدد من الصفحات سريعاً مع الاحتفاظ بالبنية الهرمية لتلك الصفحات كما هي موجودة عند المخدم. بمعنى طريقة توضعها ضمن المجلدات و المجلـدات الفرعيـة، و طريقة ارتباطها ببعضها.

**٣-٥-١المبدأ الرئيسي لتطبيق الزحف:**

يبدأ عمل الزاحف من مجموعة معطاة من الوصلات (عناوين) URL، و مع التقدم في العمل يبحـث ضمن هذه الوصلات عن وصلات URL جديدة، تسمى وصلات خارجة (out links)، و من ثم يجلب تلـك الصفحات بدورها ضمن حلقة لا نهائية.

و هذا يعني أن عمل الزاحف "نظرياً" لا ينتهي، لأن المواقع لا توجد بمفردها أو مستقلة عن غيرهـا وإنما هي مترابطة ببعض بشكل أو بآخر، و على هذا لن يتوقف عمل الزاحف إلا إذا رأى المشرف أنه قـد جمع ما يكفي من الصفحات.

يجب على الزاحف أن يجلب الصفحات بشكل تفرعي، أي يجب جلب أكثر من صفحة في نفس الدورة من عمل الزاحف. السبب في توزيع و تفريع تطبيق الزاحف هو تعويض التأخيرات الزمنية الناتجة عن:

- تحويل اسم المضيف (hostname) في وصلة URL إلى عنوان IP باستخدام DNS.
- الاتصال بالمخدم باستعمال socket و إرسال الطلب.
- استقبال صفحة الإجابة.

عندما يتم الانتهاء من جلب الصفحة كاملة، يُنسخ محتواها إلى مفهرس نـصي. وجميـع الوصـلات الموجودة ضمنها تنسخ إلى وحدة للاحتفاظ بها إلى حين استدعائها.

**٣-٥ – ٢ بنية الزاحف Crawler Architecture:**

يتكون الزاحف من الوحدات التالية:

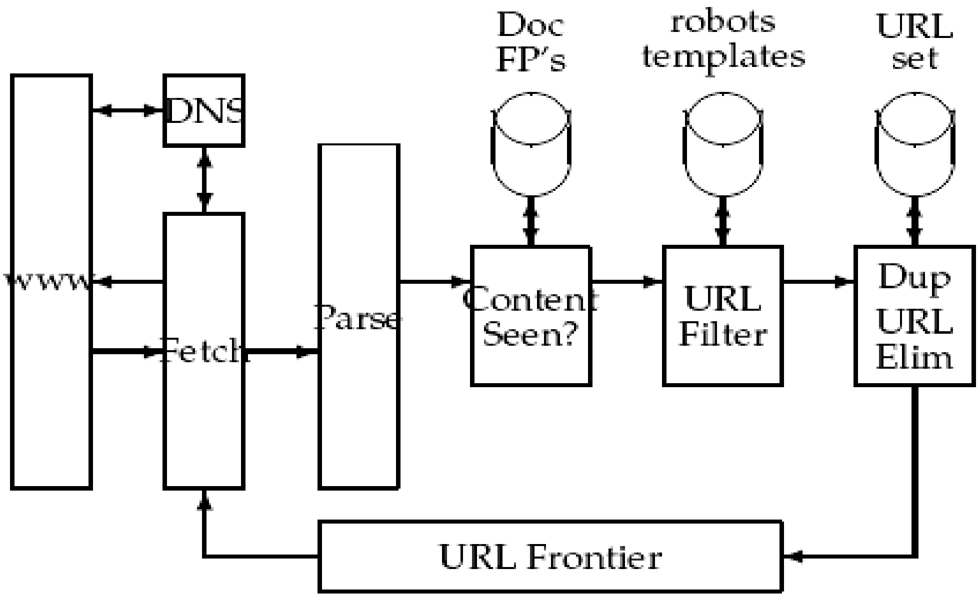
١. وحدة URL، تحتوي على الوصلات التي تنتظر الجلب في دورة الجلب هذه.
٢. وحدة تحويل DNS، تحدد مخدم الويب الذي ستجلب منه وصلات URL و صفحاتها.

٣. وحدة الجلب، تجلب صفحة الويب الخاصة بوصلة URL.

٤. وحدة (parse)، تنسخ مجموعة الوصلات من الصفحة التي تم جلبها.

٥. وحدة تحدد فيما إذا تم جلب الصفحة الخاصة بالوصلة المنسوخة أم لا.

يبين الشكل كيفية ترابط الوحدات المكونة للزاحف.




الشكل (٣-٢) بنية الزاحف

**٣-٥-٣ أساسيات الزاحف Crawling Basic :**

إن مهمة Crawler هي جلب صفحات الويب في نفس الوقت عن طريق URL(Uniform Resource Locator) حيث يبدأ العمل من URL معطى ثم يقوم بمسح باقي Link لجلب URL أخرى.

و لكن يوجد تأخير زمني أثناء عملية الجلب نتيجة تراكب عدة عمليات :

١ - تحليل عنوان المضيف URL إلى IP باستخدام DNS .

٢ - فتح قناة اتصال و ربطها وإرسال الطلب .

٣ - استلام الصفحة المطلوبة.

إن دورة الحياة الموضحة سابقاً يتم قيادتها عن طريق Thread منطقي و هو ليس Thread خاصاً بنظام التشغيل لكنه مخصص و مبرمج لتقديم فعالية أكبر .

### ٣-٥-٤ الجلب المتعدد على التوازي Multiple Concurrent Fetch :

يحتاج crawler إلى فتح عدة قنوات اتصال لمخدمات HTTP مختلفة في نفس الوقت وذلك بـــسبب أن عملية تحميل واحدة تستهلك عدة ثوان.

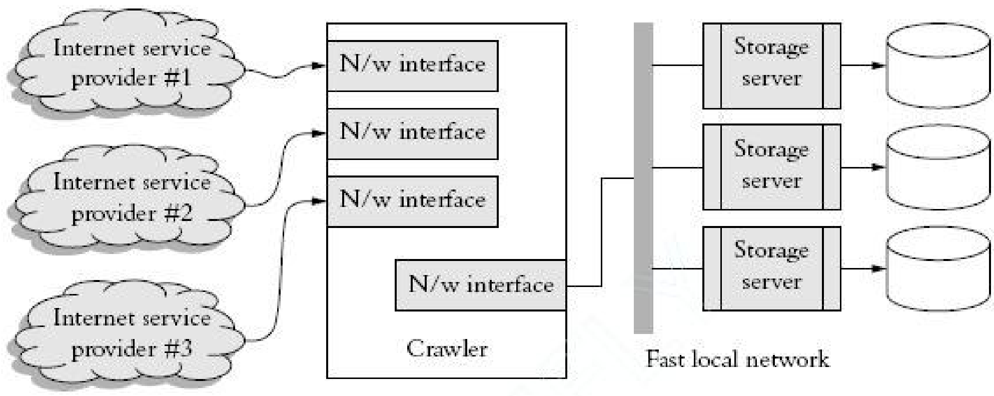
يوجد طريقتان لإدارة عملية الاتصالات المتعددة على التوازي :

١- المسارات المتعددة Multithreading.

٢- الممرات غير المغلقة مع ماسك الحدث no blocking sockets with event handler.

إن استخدام حاسب بعدة معالجات لن يكون ذا نفع كبير وذلك بسبب محدودية الشبكة والأقراص .

كما يوضح الشكل التالي آلية عمل الزاحف:




الشكل (٣-٣)

# الفصل الرابع

## التنقيب في المعطيات

١. مقدمة عن التنقيب في المعطيات.

٢. مهام التنقيب في المعطيات.

٣. وظائف التنقيب في المعطيات.

٤. مسألة تصنيف النصوص.

٥. سيناريو التعلم بإشراف.

## ٤- ١مقدمة عن التنقيب في المعطيات Introduction to Data Mining:

رافق ظهور الحاسب انتشار ظاهرة سمع بها الكثيرون وهي ظاهرة انفجار البيانات، أي بيانات هائلة متعددة الأشكال والبنى. و جاء الرد على هذه الظاهرة بابتكار أنظمة تخزين المعطيات بشكل مهيكل سُميت بقواعد المعطيات، و تعددت أشكال نظم إدارة هذه القواعد حتى أصبحت علماً قائماً بذاته. و من أكثر أنواعها انتشاراً نظم إدارة قواعد المعطيات العلائقية

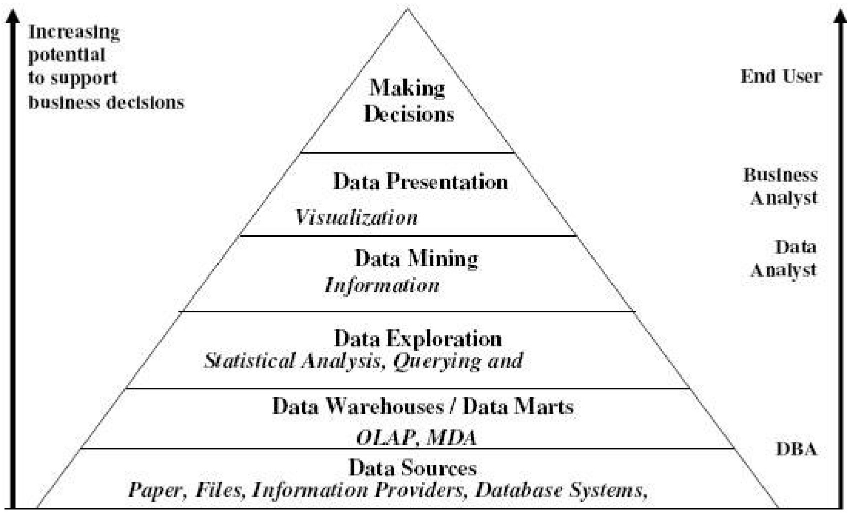
(Relational Database Management Systems) أو اختصاراً بـRDBMS.

كما ظهر مؤخراً أنواع جديدة منها نظم إدارة قواعد المعطيات غرضية التوجه (OOPDBMS). و لكن مع مرور الزمن بدأت تظهر مشكلة جديدة وهي ازدياد حجم قواعد المعطيات تلك و ازدياد ضخامتها، إذ بتنا نتحدث عن قواعد بيانات تحوي بيانات جميع مواطني دولة معينة مثلاً.

و هذا قادنا إلى حالة تتمثل في الغنى في المعطيات و الفقر في المعلومات

(we are data rich, but information poor).

- الآن كيف يمكننا الحصول على ما نتوقعه من المعلومات الدقيقة من هذا الكم الهائل من المعطيات؟جاء رداً على السؤال السابق، التنقيب عن المعرفة في المعطيات، حيث الغاية من التنقيب هي الحصول على معرفة (نماذج معرفية knowledge patterns).




الشكل (٤-١) من المعطيات إلى المعرفة

كما يطلق على هذه العملية استكشاف المعرفة (knowledge discovery).

**تعريف التنقيب في المعطيات (Data mining):**

عملية استخلاص أو اشتقاق المعلومات الضمنية (المخفية)، غير المعروفة سابقاً، و التي من المرجح أن تكون على قدر كبير من الأهمية وذلك من خلال الحصول عليها من المعلومات التي تتواجد في قواعد المعطيات.

عادة يتم استخدام المصطلحين التنقيب في المعطيات واستكشاف المعرفة في قواعد المعطيات كمترادفين، ولكن في الحقيقة التنقيب في المعطيات هو جزء فقط من عملية استكشاف المعرفة.

- ما هي نوعية المعطيات التي يمكن التنقيب فيها؟

عملية التنقيب يجب أن تكون قابلة للتطبيق على أي نوع من مستودعات المعطيات. ولكن يتضح الاختلاف هنا على مستوى الطرائق والخوارزميات حسب المجال قيد الدراسة.

ما يهم هنا من مختلف المجالات هو شبكة الانترنت من المعروف أن شبكة الويب هي أكثر مستودعات المعطيات تغيراً (ديناميكية) كما أنها تتصف بعدم التجانس.

**٤-٢مهام التنقيب في المعطيات Data mining tasks:**

هناك مهمتان أساسيتان:

١. التنقيب الوصفي في المعطيات Descriptive data mining.

٢. التنقيب التنبؤي في المعطيات Predictive data mining.

يصف التنقيب الوصفي الخصائص العامة للبيانات الموجودة. بينما يحاول التنقيب التنبؤي التنبؤ اعتماداً على الاستدلال من المعطيات المتوفرة.

**٤-٣ وظائف التنقيب في المعطيات:**

١. تحليل الترابط Association Analysis.

٢. التصنيف Classification.

٣. التوصيف Characterization.

٤. تحليل الشذوذ Outlier Analysis.

٥. التنبؤ Preduction.

٦. و غيرها...

**٤-٤التنقيب في المعطيات النصية على الويب Text Mining on The Web:**

**تعريف التنقيب في النص (text mining) :**

تحليل تجمعات المستندات الهائلة من أجل اشتقاق النماذج المعرفية المخفية من المعطيات النصية.ولكن من وجهة نظر خبراء شركة أوراكل ®Oracle فإن التنقيب النصي هو:

استخدام معالجة اللغات الطبيعيـــة ( Natural language processing NLP) والتقنيـــات الإحـــصائية / الاحتمالية لإيجاد الكتل والنماذج المعرفية في التجمعات النصية.

حيث الكتلة (nugget): وحدة معرفية جديدة غير معروفة.

بينما النموذج (pattern): ورود نموذج معرفي سبق التعامل معه (معروف).

بطبيعة الحال يمكن تخيل التنقيب النصي توسعة مباشرة للتنقيب في المعطيات.

**٤ -٥ أنواع التنقيب النصي Types of text mining:**

١. تحليل الارتباط المبني على كلمات البحث.

Keyword – based association analysis.

٢. التصنيف الآلي للمستندات.

Automatic document classification.

٣. اكتشاف التماثل.

Similarity detection.

٤. تحليل الوصلات.

Link analysis.

٥. تحليل التتابعية.

Sequence analysis.

٦. تحليل النصوص التشعبية.

Hypertext analysis.

**٤- ٦ تصنيف النصوص و التعلم بإشراف Text classification & supervised learning:**

**الهدف:** التصنيف الآلي (الأوتوماتيكي) لعدد كبير من المستندات النصية المتوافرة على شبكة الويب (صفحات الويب – البريد الالكتروني...).

هذه العملية تختلف عن عملية التصنيف في قواعد المعطيات العلائقية، حيث إن المستندات هي بُنى غير مهيكلة اعتماداً على أزواج (الخاصية – القيمة) كما في قواعد المعطيات.

وعلى اعتبار أن عملية تصنيف المستندات مرتبطة بكلٍ من نظم استرجاع المعلومات الكلاسيكية (classical IR) و نظم التنقيب في محتوى الويب.

لا يوجد عملياً أي نظام معلوماتي مؤتمت (أو غير مؤتمت) يستطيع أرشفة وتصنيف جميع المستندات النصية المتوافرة على الويب. ولذلك تلجأ معظم النظم (محركات بحث، نظم تجميع، وغيرها...) إلى أرشفة و تصنيف جزء من الويب بحيث يكون ثابتاً أي ستاتيكي.

**<u>تعريف صفحات الويب الثابتة (الستاتيكية):</u>**

الصفحات التي لا يتغير محتواها بين المشاهدات المختلفة لهذه الصفحة من قبل المستخدمين المختلفين.

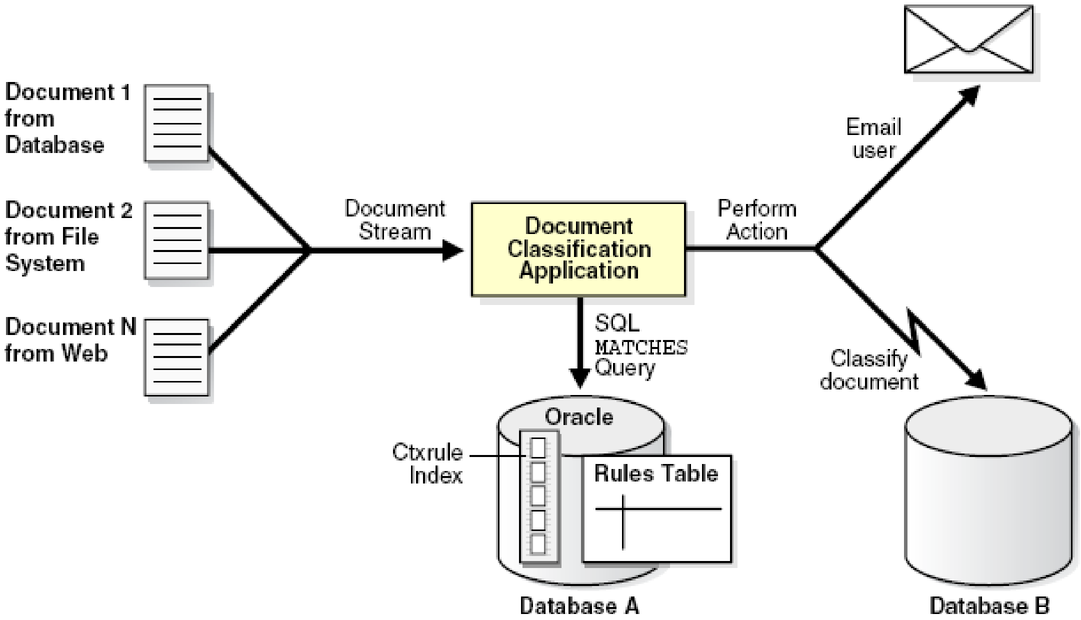
**<u>تعريف صورة الويب المؤرشفة (web graph):</u>**

مجموعة من صفحات الويب الثابتة، بالإضافة إلى الارتباطات التشعبية فيما بينها (جزء من الويب).

– ما السبيل إلى تصنيف صورة الويب المؤرشفة؟

هنا يأتي دور تقنيات التنقيب في المعطيات نصية كانت أم غيرها. و كما ذُكر سابقاً، فإن الويب كمحتوى يتكون ليس فقط من المستندات النصية، وإنما من مستندات الوسائط المتعددة أيضاً كالصور، ملفات الصوت، ملفات الفيديو، وغيرها...

والشكل التالي يبين بنية تصنيف في الويب :




الشكل (٤-٢) نظرة عامة على بنية تطبيق التصنيف.

- و لكن ما المعنى الدقيق للتصنيف الآلي؟

ينطوي التصنيف الآلي ضمن ما يسمى التعلُّم الآلي مع الإشراف (Supervised Machine Learning)، والذي يهدف إلى تحديد ما إذا كان كل مستند في تجمع المستندات ينتمي إلى صف (فئة) أو أكثر من الصفوف المحددة مسبقاً أم لا.

طبعاً تحدد الصفوف بناءً على عدة اعتبارات، فقد تُحدد على سبيل المثال اعتماداً على الفكرة المحورية للمستند.

أما سبب التسمية فيعود إلى أن عملية تحديد الصفوف يتم الإشراف عليها من قبل خبير (إنسان).

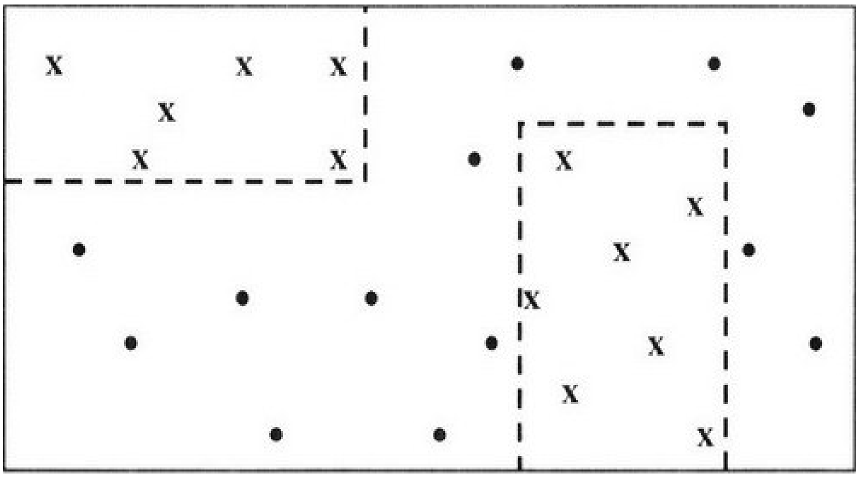
### ٤-٦-١ مسألة تصنيف النصوص The text classification problem:

يُعطى التصنيف المعتمِد على خوارزميات التعلُّم الاستقرائي كدخل مجموعةً من عينات (بيانات) تدريب تتكون من أشعة (vectors) من قيم الخواص (feature values or attribute values)، وصف مقابل لتلك المعطيات يمثل الهدف.

الهدف من عملية التعلم هو توليد نموذج تصنيف (classification model)، يُعرف على أنـــه مـــصنِّف (classifier) سيتوقع بالاعتماد على قيم الخواص المعطاة كدخل الصف الخاص بعينة اختبار لا تنتمـــي إلـــى بيانات التدريب.

**توضيح:**

تتطلب عملية التصنيف وجود أكثر من صف واحد، كل صف له مجموعة من القواعـــد و الـــشروط المميِّزة له و التي إذا ما حققها مستند ما سينتمي ذلك المستند إلى الصف المحدد.



الشكل (٣،٤) تصنيف للمستندات ثنائي البعد

في تصنيف النصوص، يُعطى وصف لمثيل x ينتمي إلى X، حيث X هو فضاء الأمثلة.

ومجموعة ثابتة من الصفوف (الفئات) $C = \{c١, c٢, ..., cj\}$، الأمثلة = مستندات. حيث أن الـــصفوف مُعرَّفة من قبل الخبراء.

ليكن لدينا مجموعة أو أكثر من مستندات التدريب (training documents)، ومن أجل كل صـــف مـــن مجموعة الصفوف j نريد أن يتم تعلم تابع تصنيف (classification function) $f$، يقوم بإسقاط الأمثلة علـــى الصفوف.

$$f: X \rightarrow C$$

وبالتالي تنحصر مهمة الخبير في تحديد فئات المستندات والقيام بوظيفة المعلم للتوجيه والإشراف على عملية التعلم.

تعريف بيانات التدريب (training data):

مجموعة من المعطيات منتقاة بعناية تزودنا ببعض الأمثلة النموذجية من أجل كل صف وبالتالي نتعلم التابع $f$.

ثم بعد عملية التعلم نستطيع تطبيق $f$ على بيانات الاختبار (test data).

**٤-٦-٢ سيناريو التعلم بإشراف The supervised learning scenario:**

يتلقى خبراء وعلماء المكتبات تدريبات مكثفة حتى يصبحوا قادرين على عنونة المنشورات المختلفة باستخدام عناوين مواضيع صحيحة وشاملة.

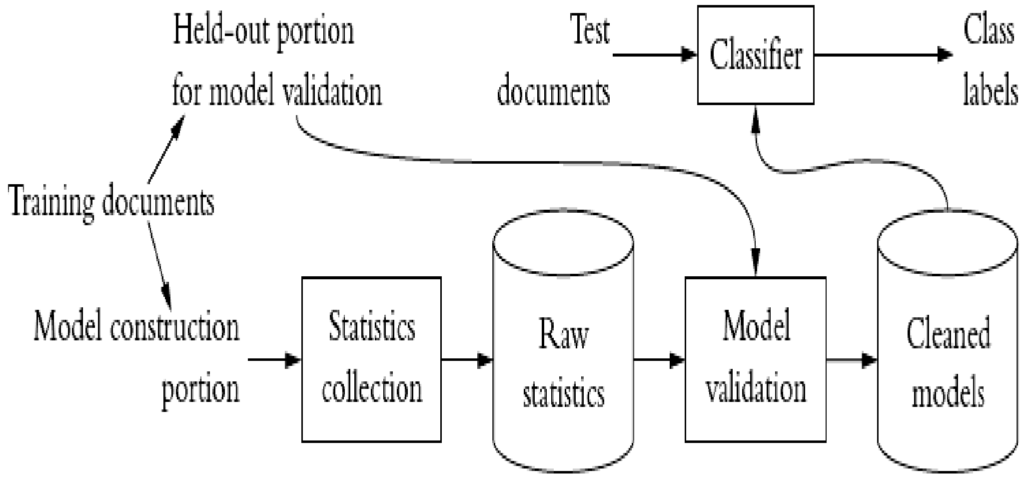
- السؤال المطروح الآن، هل يستطيع الحاسب أن يمتلك ولو جزءاً بسيطاً من تلك الخبرة؟

هذا السؤال بالتأكيد مهم وذلك من وجهة نظر خبراء الويب، بسبب الحجم المتنامي للويب بالإضافة إلى نقص التحكم بالمحتوى والمضمون، أي غياب الرقابة والتحرير.

ضمن هذا السياق، يقوم المصنِّف (classifier) (أو قد يطلق عليه المتعلم learner) باستقبال بيانات التدريب بدون علامات (labels) تدل على أنها تنتمي إلى صف معين أو فئة محددة، وبالتالي نستخدم هذه المعطيات لتدريب خوارزمية التعلم.

حالما ينهي المصنف التدريب، يعطى مجموعة من المعطيات غير المُعلَّمة تسمى بيانات اختبار (test data)، ويجب عليه عندئذٍ أن يصنفها.

ولتبيان هذا السيناريو أورد الشكل التالي.




الشكل (٣,٤) سيناريو التعلم بإشراف

# الفصل الخامس

**Classification Algorithms**

**خوارزميات التصنيف:**

٤- خوارزمية التعلم الذاتي Rule١.

٤- خوارزمية Rule٣.

٤- خوارزمية Rule٣ Plus.

٤- خوارزمية Rule٥.

٤- خوارزمية شجرة التصنيف والارتداد CART.

٤- خوارزمية بناء الشجرة C٤,٥ .

٤- تعديل الخوارزمية C٤,٥.

**٥-١ مقدمة:**

التعلم الذاتي (inductive learning)[١٨،٢٧] يعني استنتاج قواعد عامة من قواعد بيانات ضخمة.

سنركز في هذا الفصل على التعلم الذاتي وخوارزميات التصنيف لما لعملية التصنيف من دور هام في عملية جلب واسترجاع المعطيات واستنباط المعرفة.

**٥-٢ من أهم تطبيقات التعلم الذاتي[٢٧]:**

- تمييز النماذج (pattern recognition).
- تمييز الأوراق النقدية (banknote recognition).
- تمييز التوقيع (signature recognition).
- تمييز الباركود (barcode recognition).

**٥-٣ أنواع خوارزميات التعلم الذاتي[٢٩]:**

أ- خوارزميات أشجار القرار(decision tree): تقوم بإنتاج شجرة قرار ثم عبور هذه الشجرة لإنتاج القواعد مثال خوارزمية [Quinlan][٨،١٨]ID٣ ، CART[٢]

ب- خوارزميات التغطية (covering algorithm): تقوم بإنتاج القواعد دون إيجاد شجرة قرار مثل خوارزمية [Pham, Aksoy, Bigot][١٩،٢١]RULES AQ[٢٩].

هناك العديد من الخوارزميات المعروفة في عملية التصنيف :

١- خوارزمية Rule١.

٢- خوارزمية Rule٣.

٣- خوارزمية Rule٣ Plus.

٤- خوارزمية Rule٥.

٥- خوارزمية شجرة التصنيف والارتداد CART.

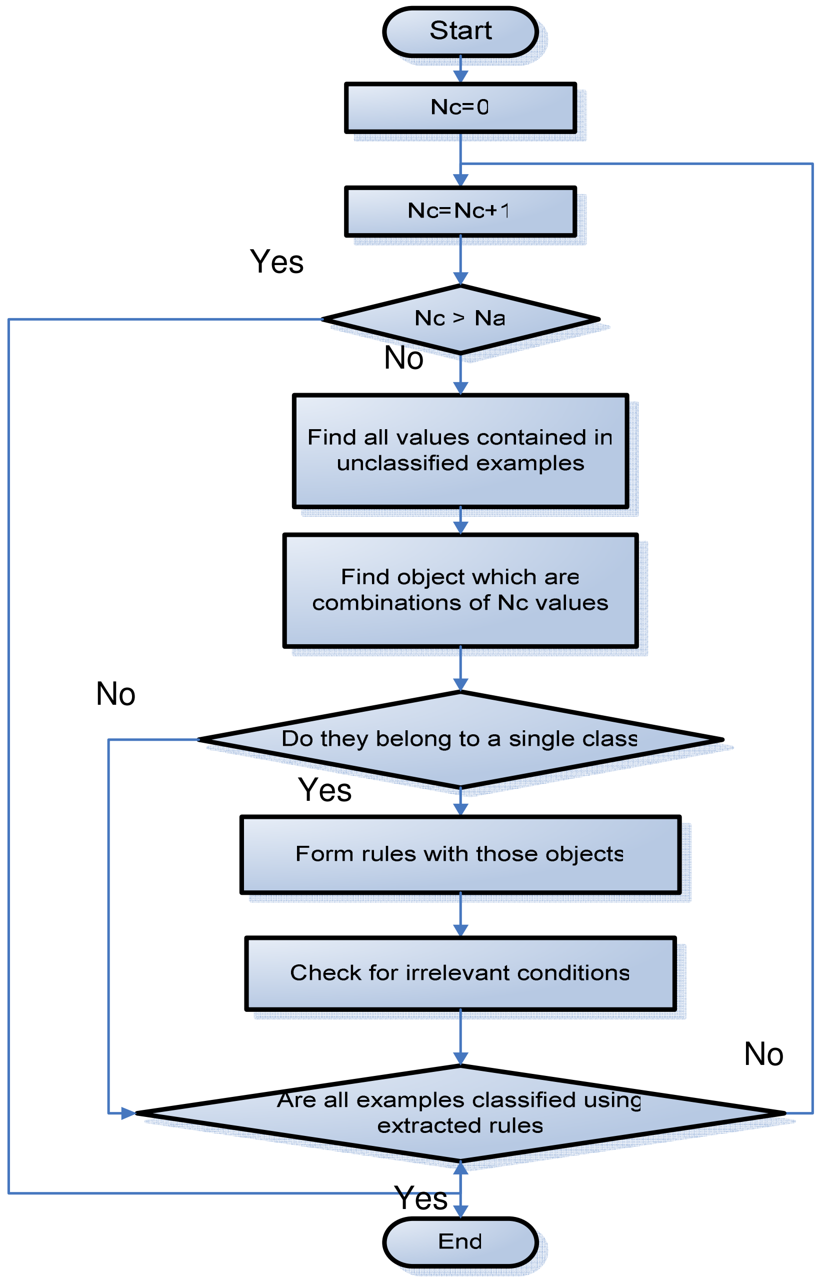
٦- خوارزمية بناء الشجرة C٤,٥ .

ومن خلال هذا الفصل سيتم أيضاً مقارنة كل من خوارزمية CART مع خوارزمية C٤,٥ ومن ثم مقارنة خوارزمية C٤,٥ مع تعديلاتها المقترحة.

### ٥-٣-١ خوارزمية التعلم الذاتي RULES-1:

هي طريقة بسيطة لاستنتاج قوانين تصنيف من مجموعة من النماذج examples التي تنتمي إلى أكثر من صنف class، حيث يتم وصف النموذج من خلال عدد من الصفات attributes وهذه الصفات يجب أن تكون قيماً اسمية (nominal ) وهو من عيوب هذه الخوارزمية.

يوضح الشكل (٥-١) هذه الخوارزمية.




الشكل(٥،١)

حيث يقوم مبدأ الخوارزمية على البدء بعدد شروط Nc=٠ ومن ثم البدء بتشكيل القاعدة من شرط واحد في كل مرة حتى الوصول إلى قاعدة متماسكة أو أن نتجاوز عدد الصفات، حيث يتم البحث من أجل كل قاعدة مقترحة عن الأمثلة التي تنتمي لصنف واحد عندها يتم تشكيل القاعدة المحققة لهذه الأمثلة ومن ثم يتم فحص فيما إذا بقي أمثلة لم يتم تصنيفها عندها يتم تكرار الخوارزمية بزيادة شرط آخر للقاعدة حتى الوصول لعدد شروط في القاعدة أكبر من عدد الصفات وإلا تتوقف عمل الخوارزمية.

**٥-٣-٢ خوارزمية التعلم الذاتي RULES-٣:**

**RULES-٣ Inductive Learning Algorithm:**

هي طريقة بسيطة لاستنتاج قوانين التصنيف من مجموعة من النماذج examples التي تنتمي إلى أكثر من صنف class حيث يتم وصف النموذج من خلال عدد من الصفات attributes وهذه الصفات يمكن أن تحمل قيماً رقمية (numerical ) أو قيماً اسمية (nominal ) . على سبيل المثال صفة الطول يمكن أن تحمل قيماً اسمية مثل ( short , medium , long ) ويمكن أيضاً أن تحمل قيماً رقمية ضمن المجال ( ١٠ ,١٠- ) مثلاً. الثنائية صفة – قيمة ( attribute – value ) تشكل شرطاً condition في القاعدة (rule ) .

بفرض أن عدد الصفات هو Na عندها فإن القاعدة يمكن أن تحتوي على عدد من الشروط يتراوح بين الواحد و Na .

يتم استنتاج القواعد بأخذ نموذج من النماذج في كل مرة ، حيث نقوم بأخذ القواعد التي تتألف من شرط واحد و ننظر إلى باقي النماذج فإذا توافقت باقي النماذج مع القاعدة التي فرضناها عندها هذه القاعدة تصلح لأن تكون قاعدة صحيحة (متماسكة) وفي حال وجود أكثر من قاعدة صحيحة نأخذ القاعدة التي تصنف العدد الأكبر من النماذج ، وفي حال عدم وجود أي قاعدة متماسكة مؤلفة من شرط واحد عندها نأخذ القواعد التي تتألف من شرطين وهكذا حتى الوصول إلى قاعدة متماسكة تغطي النموذج المختار ويمكن أن تغطي القاعدة المستنتجة نماذج أخرى، في أسوا الأحوال تكون القاعدة المستنتجة تحوي جميع الشروط ، بعد استنتاج

القاعدة يتم حذف جميع النماذج المصنفة (classified examples) ونأخذ نموذج آخر ونستمر في ذلك حتى يتم الانتهاء من جميع النماذج .

يمكن تلخيص هذه الخوارزمية على الشكل التالي :

الخطوة١ : تحديد المجالات للصفات التي تحمل قيماً رقمية. وتحديد أسماء لهذه المجـــالات (طـــرح القيمـــة الصغرى من القيمة العظمى للصفة وتقسيم الناتج على عدد المجالات الذي يتم تحديده من قبل المستخدم).

الخطوة٢: تحديد عدد الشروط الرئيسية لكل قاعدة (Ncmin).

الخطوة٣: أخذ نموذج غير مصنف.

الخطوة٤: Nc=Ncmin-١ .

الخطوة٥: إذا كان Nc<Na عندها Nc=Nc+١ .

الخطوة ٦: أخذ كل القيم أو الأسماء المتبقية في النموذج.

الخطوة٧: تشكيل العناصر والتي تشكل تجميع من قيم Nc أو الأسماء المأخوذة من القيم أو الأسماء التي تـــم الحصول عليها في الخطوة٦.

الخطوة٨: إذا وجد على الأقل عنصر واحد ينتمي إلى صنف وحيد عندها يتم تشكل القاعدة من هذا العنصر، وإلا يتم الانتقال إلى الخطوة ٥.

الخطوة٩: اختيار القاعدة التي تصنف العدد الأكبر من النماذج.

الخطوة١٠: حذف جميع النماذج التي تصنفها القاعدة المختارة.

الخطوة١١: إذا لم يبق أي نماذج غير مصنفة توقف وإلا فالانتقال إلى الخطوة٣.

حيث أن Nc يمثل عدد الشروط في كل قاعدة .

لتوضيح هذه الخوارزمية سيتم تمرير المثال الموضح في الجدول (٥-١).

الجدول(١،٥)

| Example no | CarSize | Fuel | MaxSpeed | Acceleration |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| ١ | Large | Diesel | High | Good |
| ٢ | Large | Propane | High | Good |
| ٣ | Compact | Petrol | High | Excellent |
| ٤ | Large | Petrol | High | Excellent |
| ٥ | Medium | Diesel | Low | Good |
| ٦ | Compact | Petrol | Low | Good |
| ٧ | Medium | Petrol | Average | Excellent |
| ٨ | Medium | Diesel | Average | Poor |

أ- يتم أخذ النموذج رقم ١

ب- في أول دورة للحلقة نقوم بإنتاج قواعد تحوي شرطاً واحداً فقط.

IF CarSize = Large THEN Acceleration = Good

ت- نلاحظ أن النموذج رقم ٤ لا يصنف من قبل هذه القاعدة ومنه فإن هذه القاعدة لا تصلح

IF Fuel = Diesel THEN Acceleration = Good

ث- نلاحظ أن النموذج رقم ٨ لا يصنف من قبل هذه القاعدة ومنه فإن هذه القاعدة لا تصلح

IF MaxSpeed = High THEN Acceleration = Good

ج- نلاحظ أن النموذجين رقم ٣ والنموذج رقم ٤ لا يصنفان من قبل هذه القاعدة ومنه فإن هـــذه القاعدة لا تصلح .

ح- لم نحصل على أي قاعدة صحيحة لذلك نقوم بإنتاج قواعد تحوي على شرطين .

IF CarSize = Large AND Fuel= Diesel THEN Acceleration=Good

خ- هذه القاعدة تصنف نموذجاً واحداً فقط ولا يوجد أي نموذج غير مصنف من هذه القاعدة.

IF Fuel =Diesel AND MaxSpeed=High THEN Acceleration=Good

د- هذه القاعدة تصنف نموذجاً واحداً فقط ولا يوجد أي نموذج غير مصنف من هذه القاعدة.

ذ- نختارعشوائياً إحدى القاعدتين ، وهنا اخترنا القاعدة الأولى ، ثم نحذف النموذج الأول.

ر- نأخذ النموذج رقم ٢ ونقوم بتوليد قواعد تحتوي على نموذج واحد فقط

IF CarSize = Large THEN Acceleration = Good

ز- نلاحظ أن النموذج رقم ٨ لا يصنف من قبل هذه القاعدة ومنه فإن هذه القاعدة لا تصلح.

IF Fuel = Propane THEN Acceleration = Good

س- تصنف هذه القاعدة نموذجاً واحداً فقط ولا يوجد أي نموذج غير مصنف من هذه القاعدة ومنه فإن هذه القاعدة تصلح .

IF MaxSpeed = High THEN Acceleration = Good

ش- نلاحظ أن النموذجين رقم ٣ والنموذج رقم ٤ لا يصنفان من قبل هذه القاعدة ومنه فإن هذه القاعدة لا تصلح .

ص- ومنه القاعدة التي يتم توليدها من هذا النموذج هي :

IF Fuel = Propane THEN Acceleration = Good

ض- نحذف النموذج الثاني .

ط- نكرر العملية حتى الانتهاء من جميع النماذج ، حيث نحصل في النهاية على القواعد التالية:

١.IF CarSize=Large AND Fuel=Diesel THEN Acceleration = Good

٢.IF Fuel=Propane THEN Acceleration = Good

٣.IF Fuel=Petrol AND MaxSpeed=High THEN Acceleration = Excellent

٤.IF MaxSpeed=Low THEN Acceleration = Good

٥.IF CarSize=Medium AND Fuel=Petrol THEN Acceleration = Excellent

٦.IF Fuel=Diesel AND MaxSpeed=Average THEN Acceleration = Poor

$n_c$: عدد الشروط.

$N_a$: عدد الصفات.

**٥-٣-٢-١ نقاط القوة:**

١- تتعامل خوارزمية Rule٣ مع القيم الأسمية والقيم الرقمية.

**٥-٣-٢-٢ نقاط الضعف:**

١- تحتاج هذه الخوارزمية لتحديد مجال القيم الرقمية حتى تستطيع التعامل معها وبالتالي هي عملية طويلة.

٢- تقوم الخوارزمية بأخذ الشروط بشكل متسلسل دون أفضلية أو اعتماد مبدأ معين للاختيار.

٣- تختار القانون الذي يصنف العدد الأكبر من النماذج.

## ٥-٣-٣ خوارزمية التعلم الذاتي RULES-٣ PLUS:

**RULES-٣ PLUS Inductive Learning Algorithm:**

مبدأ العمل : نأخذ نموذجاً من Training Data ونسمي هذا النموذج بـــالنموذج البـــذرة (Seed Example) وعادة يكون أول نموذج غير معلم (unmarked example) في الـــ Data Training من النموذج البـــذرة SE نقوم بإنشاء قاعدة عامة if nocond then class

ثم نقوم بتخصيص هذه القاعدة حتى الوصول إلى قاعدة متماسكة حيث نبدأ أولاً بإضافة شرط واحد للقاعدة (نأخذ جميع الاحتمالات ). نحسب مقياس H لكل قاعدة ( تتألف من شرط واحد) ونفحص ما إذا كانت توجد ضمن هذه القواعد المشتقة قاعدة متماسكة (consistent) أي n=٠ ( عدد النماذج التي تنتمي إلى Class الهدف ولا تغطيها القاعدة) في حال وجود أكثر من قاعدة متماسكة نختار القاعدة التي لها مقياس H أكبر ونتوقف . في حال عدم وجود أي قاعدة متماسكة ننتقل إلى مناقشة القواعد التي تحتوي على شرطين حيث نأخـــذ مـــن القواعد السابقة (القواعد ذات الشرط الواحد) القاعدة التي لها أكبر مقياس H ونقوم بإضافة الشرط الثاني لهذه القاعدة حتى الوصول إلى قاعدة متماسكة في حال عدم الحصول على قاعدة متماسكة عندها نأخذ القاعدة ذات الشرط الواحد التي تليها بمقياس H ونحاول تخصيصها ( إضافة شرط ثان ) وذلك حتى الوصول إلى قاعدة متماسكة . وفي حال الانتهاء من تخصيص جميع القواعد ذات الشرط الواحد عن طريق إضافة شرط ثـــاني ولم نحصل على قاعدة متماسكة عندها ننتقل إلى إضافة شرط ثالث وهكذا . عند الحـــصول علـــى قاعـــدة متماسكة نضيفها إلى مجموعة القواعد ونعلم جميع النماذج التي تصنفها هذه القاعدة ثم ننتقل إلى SE جديد ( أول نموذج غير معلم ) وهكذا حتى انتهاء جميع النماذج .الشكل(٢،٤) يبين هذه الخوارزمية.

(step ١)

Quantize attributes that have nominal values.

(step ٢)

Take a seed example not covered by the rule set formed so far and form array AttributesAndValues.

(step ٣)

Initialise PartialRules with the most general rule (a rule with no conditions) and set *nc* = ٠.

(step ٤)

**If** *nc* < *Na* **Then**

*nc* = *nc* + ١

Initialise T_PartialRules to empty and set *n* = ٠.

**Else**

Take the example itself as a rule and go to step (٧).

(step ٥)

**Do**

*n* = *n* +١

Form a list of rules (T_Rules), the elements of which are combinations of rule *n* in PartialRules with conditions from AttributesAndValues that differ from the conditions already included in rule *n*

(the number of elements in T_Rules is: *Na* - *nc*) and set *l* = ٠.

**Do**

*l* = *l* + ١

Compute the *H* measure for rule *l* in T_Rules.

**If** number of rules in T_PartialRules < *w* **Then**

Store rule *l* into T_PartialRules.

**Else**

**If** the *H* measure of rule *l* is higher than the *H* measure of any rule in T_PartialRules **Then**

Replace the rule having the lowest *H* measure in T_PartialRules with rule *l*.

**Else** Discard rule *l*.

**While** *l* < *Na* - *nc*

**While** PartialRules ≠ ∅ **AND** *n* < *w*

(step ٦)

**If** there are consistent rules in T_PartialRules **Then**

Add to the rule list the consistent rule that has the highest *H* measure and discard the others.

step (٧)

Mark the examples covered by this rule and go to

**Else** Copy T_PartialRules into PartialRules and go to step (٤).

**If** there are no more examples not covered **Then** Stop.

**Else** Go to step (٢).

الشكل(٢،٥)

$n_c$**:** عدد الشروط.

$N_a$**:** عدد الصفات.

***w***:عدد القواعد المخزنة في القائمة الجزئية ***PartialRules*** (القيمة العظمى يتم تحديدها من قبل المستخدم).

***T_PartialRules***: قائمة مؤقتة من القواعد حجمها يساوي حجم القائمة الجزئية.

***T_Rules***: قائمة مؤقتة من القواعد.

**٥-٣-٣-١ نقاط القوة** : الجديد الذي أضافته هذه الخوارزمية على RULES-٣ هي استخدامها مقياس H . النقطة الثانية هي أن عملية توليد القواعد في RULES-٣ كانت تبدأ بأخذ الشروط بشكل متسلسل دون الأخذ بعين الاعتبار الأفضلية أما خوارزمية RULES-٣ PLUS فتأخذ شرط واحد ( جميع الاحتمالات ) وفي حال عدم الحصول على قاعدة متماسكة ننتقل إلى شرطين ولكن نبدأ هنا بتخصيص القاعدة ذات الشرط الواحد التي تملك مقياس H أكبر وهذا يسرع في عملية استنتاج القوانين .

**٥-٣-٣-٢ نقاط الضعف** : المشكلة الرئيسية في هذه الخوارزمية أنها تتقيد بالمعطيات Training Data تقيّداً كاملاً حيث أنها تغطي جميع النماذج الموجبة ( النماذج التي تنتمي إلى Class الهدف ) ولا تغطي أي نموذج ( النماذج التي لا تنتمي إلى Class الهدف) ندعو هذا بـــ both consistent and complete وهذا غير مفيد في حالة المعطيات التي تحوي على معلومات غير مفيدة (noisy data) حيث أن الخوارزمية تتقيد بهذه المعطيات غير المفيدة وتولد لها قواعد (قواعد غير مفيدة عملياً) وهذا يؤدي إلى الحصول على عدد كبير من القواعد . المشكلة الثانية أنها تحتاج إلى عملية تقطيع (discretisation) للقيم المستمرة continues وعملية التقطيع فـــي هذه الخوارزمية هي عملية عشوائية تطبق على المعطيات بدون إجراء دراسة على هذه المعطيات قبل تطبيق التقطيع . المشكلة الثالثة أن الخوارزمية ليست تزايدية ( non incremental) حيث لا تمكن من تعديل القواعد الناتجة عند وجود نماذج جديدة .

**مثال توضيحي**: بفرض قاعدة المعطيات الموضحة في الجدول (٢،٥) والتي تحتوي على معلومات عن ثلاثة أصناف من الأزهار.

الجدول (٢،٥)

| SL | SW | PL | PW |
|---|---|---|---|
| Sepal length | Sepal width | Petal length | Petal width |
| طول السبلة | عرض السبلة | طول البتلة | عرض البتلة |

| Class[d] | SL[c] | SW[c] | PL[c] | PW[c] | Class[d] | SL[c] | SW[c] | PL[c] | PW[c] |
|---|---|---|---|---|---|---|---|---|---|
| Iris_Setosa | ٤,٩ | ٣,٠ | ١,٤ | ٠,٢ | Iris_Versicolor | ٦,٧ | ٣,٠ | ٥,٠ | ١,٧ |
| Iris_Setosa | ٤,٦ | ٣,١ | ١,٥ | ٠,٢ | Iris_Versicolor | ٥,٧ | ٢,٦ | ٣,٥ | ١,٠ |
| Iris_Setosa | ٥,٤ | ٣,٩ | ١,٧ | ٠,٤ | Iris_Versicolor | ٥,٥ | ٢,٤ | ٣,٧ | ١,٠ |
| Iris_Setosa | ٥,٠ | ٣,٤ | ١,٥ | ٠,٢ | Iris_Versicolor | ٦,٠ | ٢,٧ | ٥,١ | ١,٦ |
| Iris_Setosa | ٤,٩ | ٣,١ | ١,٥ | ٠,١ | Iris_Versicolor | ٦,٠ | ٣,٤ | ٤,٥ | ١,٦ |
| Iris_Setosa | ٤,٨ | ٣,٤ | ١,٦ | ٠,٢ | Iris_Versicolor | ٦,٣ | ٢,٣ | ٤,٤ | ١,٣ |
| Iris_Setosa | ٤,٣ | ٣,٠ | ١,١ | ٠,١ | Iris_Versicolor | ٥,٥ | ٢,٥ | ٤,٠ | ١,٣ |
| Iris_Setosa | ٥,٧ | ٤,٤ | ١,٥ | ٠,٤ | Iris_Versicolor | ٦,١ | ٣,٠ | ٤,٦ | ١,٤ |
| Iris_Setosa | ٥,١ | ٣,٥ | ١,٤ | ٠,٣ | Iris_Versicolor | ٥,٠ | ٢,٣ | ٣,٣ | ١,٠ |
| Iris_Setosa | ٥,١ | ٣,٨ | ١,٥ | ٠,٣ | Iris_Versicolor | ٥,٧ | ٣,٠ | ٤,٢ | ١,٢ |
| Iris_Setosa | ٥,١ | ٣,٧ | ١,٥ | ٠,٤ | Iris_Versicolor | ٥,٧ | ٢,٩ | ٤,٢ | ١,٣ |
| Iris_Setosa | ٥,١ | ٣,٣ | ١,٧ | ٠,٥ | Iris_Versicolor | ٦,٢ | ٢,٩ | ٤,٣ | ١,٣ |
| Iris_Setosa | ٥,٠ | ٣,٠ | ١,٦ | ٠,٢ | Iris_Versicolor | ٥,١ | ٢,٥ | ٣,٠ | ١,١ |
| Iris_Setosa | ٥,٢ | ٣,٥ | ١,٥ | ٠,٢ | Iris_Versicolor | ٥,٧ | ٢,٨ | ٤,١ | ١,٣ |
| Iris_Setosa | ٤,٧ | ٣,٢ | ١,٦ | ٠,٢ | Iris_Virginica | ٦,٣ | ٣,٣ | ٦,٠ | ٢,٥ |
| Iris_Setosa | ٥,٤ | ٣,٤ | ١,٥ | ٠,٤ | Iris_Virginica | ٧,١ | ٣,٠ | ٥,٩ | ٢,١ |
| Iris_Setosa | ٥,٥ | ٤,٢ | ١,٤ | ٠,٢ | Iris_Virginica | ٦,٥ | ٣,٠ | ٥,٨ | ٢,٢ |
| Iris_Setosa | ٥,٠ | ٣,٢ | ١,٢ | ٠,٢ | Iris_Virginica | ٤,٩ | ٢,٥ | ٤,٥ | ١,٧ |
| Iris_Setosa | ٤,٩ | ٣,١ | ١,٥ | ٠,١ | Iris_Virginica | ٦,٧ | ٢,٥ | ٥,٨ | ١,٨ |
| Iris_Setosa | ٥,١ | ٣,٤ | ١,٥ | ٠,٢ | Iris_Virginica | ٦,٥ | ٣,٢ | ٥,١ | ٢,٠ |
| Iris_Setosa | ٤,٥ | ٢,٣ | ١,٣ | ٠,٣ | Iris_Virginica | ٦,٨ | ٣,٠ | ٥,٥ | ٢,١ |
| Iris_Setosa | ٥,٠ | ٣,٥ | ١,٦ | ٠,٦ | Iris_Virginica | ٥,٨ | ٢,٨ | ٥,١ | ٢,٤ |
| Iris_Setosa | ٤,٨ | ٣,٠ | ١,٤ | ٠,٣ | Iris_Virginica | ٦,٥ | ٣,٠ | ٥,٥ | ١,٨ |

| | |
|---|---|
| Iris_Setosa ٥,١ ٣,٨ ١,٦ ٠,٢ | Iris_Virginica ٧,٧ ٢,٦ ٦,٩ ٢,٣ |
| Iris_Setosa ٤,٦ ٣,٢ ١,٤ ٠,٢ | Iris_Virginica ٦,٩ ٣,٢ ٥,٧ ٢,٣ |
| Iris_Setosa ٥,٣ ٣,٧ ١,٥ ٠,٢ | Iris_Virginica ٧,٧ ٢,٨ ٦,٧ ٢,٠ |
| Iris_Setosa ٥,٠ ٣,٣ ١,٤ ٠,٢ | Iris_Virginica ٦,٧ ٣,٣ ٥,٧ ٢,١ |
| Iris_Versicolor ٦,٤ ٣,٢ ٤,٥ ١,٥ | Iris_Virginica ٦,٢ ٢,٨ ٤,٨ ١,٨ |
| Iris_Versicolor ٥,٥ ٢,٣ ٤,٠ ١,٣ | Iris_Virginica ٦,٤ ٢,٨ ٥,٦ ٢,١ |
| Iris_Versicolor ٥,٧ ٢,٨ ٤,٥ ١,٣ | Iris_Virginica ٧,٤ ٢,٨ ٦,١ ١,٩ |
| Iris_Versicolor ٤,٩ ٢,٤ ٣,٣ ١,٠ | Iris_Virginica ٦,٤ ٢,٨ ٥,٦ ٢,٢ |
| Iris_Versicolor ٥,٢ ٢,٧ ٣,٩ ١,٤ | Iris_Virginica ٦,١ ٢,٦ ٥,٦ ١,٤ |
| Iris_Versicolor ٥,٩ ٣,٠ ٤,٢ ١,٥ | Iris_Virginica ٦,٣ ٣,٤ ٥,٦ ٢,٤ |
| Iris_Versicolor ٦,١ ٢,٩ ٤,٧ ١,٤ | Iris_Virginica ٦,٠ ٣,٠ ٤,٨ ١,٨ |
| Iris_Versicolor ٦,٧ ٣,١ ٤,٤ ١,٤ | Iris_Virginica ٦,٧ ٣,١ ٥,٦ ٢,٤ |
| Iris_Versicolor ٥,٨ ٢,٧ ٤,١ ١,٠ | Iris_Virginica ٥,٨ ٢,٧ ٥,١ ١,٩ |
| Iris_Versicolor ٥,٦ ٢,٥ ٣,٩ ١,١ | Iris_Virginica ٦,٧ ٣,٣ ٥,٧ ٢,٥ |
| Iris_Versicolor ٦,١ ٢,٨ ٤,٠ ١,٣ | Iris_Virginica ٦,٣ ٢,٥ ٥,٠ ١,٩ |
| Iris_Versicolor ٦,١ ٢,٨ ٤,٧ ١,٢ | Iris_Virginica ٦,٢ ٣,٤ ٥,٤ ٢,٣ |
| Iris_Versicolor ٦,٦ ٣,٠ ٤,٤ ١,٤ | Iris_Virginica ٥,٩ ٣,٠ ٥,١ ١,٨ |

نلاحظ أن هذه المعطيات تحوي على قيم رقمية لذلك نقوم بعملية تحويل هذه القيم إلى قيم متقطعة ، نسمي هذه العملية discretization . بفرض أن عدد مستويات التكميم يساوي٦ .

**الجدول (٣،٥) يوضح مستويات التكميم لكل صفة من الصفات .**

| | Min | Max | ١ | ٢ | ٣ | ٤ | ٥ | ٦ |
|---|---|---|---|---|---|---|---|---|

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| SL | ٤,٣ | ٧,٧ | ٤,٣ | ٤,٨٦٦ | ٥,٤٣٣ | ٦ | ٦,٥٦٦ | ٧,١٣ |
| SW | ٢,٣ | ٤,٤ | ٢,٣ | ٢,٦٥ | ٣ | ٣,٣٥ | ٣,٧ | ٤,٠٥ |
| PL | ١,١ | ٦,٩ | ١,١ | ٢,٠٦٦ | ٣,٠٣٣ | ٤ | ٤,٩٦٦ | ٥,٩٣٣ |
| PW | ٠,١ | ٢,٥ | ٠,١ | ٠,٥ | ٠,٩ | ١,٣ | ١,٧ | ٢,١ |

قبل البدء بتمرير الخوارزمية سيتم شرح المقياس H.

إن المقياس يتألف من حدين G والذي يعبر عن عمومية(generality) القاعدة ويعطى بالعلاقة :

$$H = A + G$$

$$G = \sqrt{\frac{p+n}{P+N}}$$

الحد A فيعبر عن دقة (accuracy) القاعدة ويعطى بالعلاقة :

$$A = 2 - 2\sqrt{\frac{p}{p+n} * \frac{p}{P+N}} - 2\sqrt{\left(1 - \frac{p}{p+n}\right)\left(1 - \frac{P}{P+N}\right)}$$

P : عدد النماذج الموجبة ( النماذج التي تنتمي إلى class الهدف)

N : عدد النماذج السالبة ( النماذج التي لا تنتمي إلى class الهدف )

p: عدد النماذج الموجبة التي تغطيها القاعدة .

n : عدد النماذج السالبة التي تغطيها القاعدة .

نأخذ أول نموذج غير معلم (unmarked example ) وهنا في مثالنا هو النموذج رقم١

| | | | | |
|---|---|---|---|---|
| A | ٢ | ٣ | ١ | ١ |

نكتب القواعد ذات الشرط الواحد

١- if SL=٢ then class=A

-نلاحظ أن النماذج ( ١ وحتى ٢٧ ) تنتمي إلى class الهدف ومنه P=٢٧

-النماذج ( ٢٨ وحتى ٨٠) لا تنتمي إلى class الهدف ومنه N=٥٣

-النماذج ( ١-٣-٤-٥-٩-١٠-١١-١٢-١٣-١٤-١٦-١٨-١٩-٢٠-٢٢-٢٤-٢٦-٢٧) تنتمي إلى class الهدف وتغطيها القاعدة ومنه p=١٨ .

-النماذج (٣١-٣٢-٤٩-٥٣-٥٨) لا تنتمي إلى class الهدف وتغطيها القاعدة ومنه n=٥.

ومنه H=٠,١١٤

٢- if SW=٣ then class=A

H= ٠,٠٠١٧ $n$=١٩ $p$=١٢ N=٥٣ P=٢٧

٣- if PL=١ then class=A

H= ٩٠,٤٨٦ $n$=٠ $p$=٢٧ N=٥٣ P=٢٧

٤- if PW=١ then class=A

H= ٠,٤٦٨٥ $n$=٠ $p$=٢٥ N=٥٣ P=٢٧

نلاحظ أن القاعدتين ٣ ، ٤ متماسكتان أي n=٠ لذلك نتوقف ونولد القاعدة التي لها مقياس H أكبر ومنه نطلق على الخرج القاعدة التالية :

R١ (if PL=١ then class=A).

### ٥-٣-٤ خوارزمية التعلم الذاتي RULES-٥:

**RULES-٥ Inductive Learning Algorithm:**

مبدأ عمل هذه الخوارزمية يعتمد أخذ نموذج من (Training data) يسمى النموذج البــذرة (Seed Example) ويدعى اختصاراً SE ثم يتم اختيار أقرب نموذج لا ينتمي إلى class الهدف للنموذج البذرة ويسمى (Closed Example) ويدعى اختصاراً CE. يتم حساب المسافة عن طريق القانون:

$$distance_E1_E2 = \sqrt{\sum_{c}\left(\frac{V_{E1}^{i} - V_{E2}^{i}}{V_{\max}^{i} - V_{\min}^{i}}\right)^{2} + \sum_{d} d_distance}$$

حيث إن:

$\sum_{c}$ مجموع الصفات المستمرة (Continues value).

$\sum_{d}$ مجموع الصفات المتقطعة (Discrete value).

$V_{E1}^{i}$ قيمة الصفة رقم i في النموذج E١.

$V_{E2}^{i}$ قيمة الصفة رقم i E٢.

$V_{\min}^{i}$ أصغر قيمة للصفة المستمرة i.

$V_{\max}^{i}$ أكبر قيمة للصفة المستمرة i.

يتم إنشاء قاعدة عامة من SE ومن ثم تخصيص هذه القاعدة بمساعدة CE.

وتستمر في عملية التخصيص حتى الوصول إلى قاعدة متماسكة (consistent) حيث في كل عملية تخصيص يتم إيجاد CE جديد ويتم التقييم باستخدام مقياس H نفسه المستخدم في الأمثلة التي تصنفها هذه القاعدة ويــتم أخذ SE جديد غير معلم وهكذا [٦].

$$V_{CE}^{I} < V_{SE}^{I} \qquad \text{if} \qquad V_{CE}^{i} < A^{i}$$

$$V_{CE}^{I} > V_{SE}^{I} \qquad \text{if} \qquad V_{CE}^{i} > A^{i}$$

ونستمر في عملية التخصيص حتى الوصول إلى قاعدة متماسكة حيث في كل عملية تخصيص يتم إيجاد CE جديد ويتم التقييم باستخدام مقياس H نفسه المستخدم في خوارزمية REULS-٣ PLUS . بعد الحصول على قاعدة متماسكة يتم تعليم جميع النماذج التي تصنفها هذه القاعدة ونأخذ SE جديد غير معلم وهكذا.

الشكل(٤،٥) يوضح هذه الخوارزمية :

Take one example uncovered by the rule set formed so far (*SE*)

*STEP1*

Initialise *PRSET* (empty list)
Form an initial rule with no conditions to classify *SEe*
Store this rule in *PRSET* and copy it into *best_rule*

*STEP2*

**WHILE** *best_rule* is not consistent **DO**
Initialise *T_PRSET* (empty list)
**FOR** each rule in *PRSET* **DO**
*rule_to specialise* = the rule taken from *PRSET*

*STEP3*

*CE* = an example misclassified by *rule_to specialise* and the closest to *SE*
**FOR** *i* = ١ to *i* = m **DO**

*STEP4*

**IF** $V_{SE}^{I} = V_{CE}^{I}$ **THEN**
*new_rule* = *rule_to_specialise*
**IF** continuous attribute **THEN**
Append the condition [ *Ai* **<** $V_{CE}^{I}$ ] to *new_rule*
**ELSE**
Append the condition [ *Ai* **=** $V_{CE}^{I}$ ] to *new_rule*

*STEP5*

**IF** *new_rule* is consistent
**AND** *new_rule* covers more uncovered examples than *best_rule* **THEN**
Replace *best_rule* with *new_rule*
**IF** *new_rule* is not consistent **THEN**
**IF** number of rules in *T_PRSET* < *PRSET*_size **THEN**
Store *new_rule* into *T_PRSET*
**ELSE**
**IF** the *new_rule* H measure is higher than the H measure of any rule in *T_PRSET* **THEN**
Replace the rule with the lowest H measure in *T_PRSET* with *new_rule*
**END FOR**
**END FOR**

*STEP6*

copy *T_PRSET* into *PRSET*
**END WHILE**
**IF** *best_rule* contains continuous attribute conditions **THEN**

Constrain their coverage to training examples

Add *best_rule* to the rule set

الشكل(٤،٥)

**٥-٣-٤-١ نقاط القوة في خوارزمية RULES-٥:**

استخدام CE له فوائد كثيرة :

١- تصغير مجال البحث حيث لا يتم فحص جميع الشروط .

٢- تقليل عدد القواعد الناتجة بشكل كبير .

٣- القيم المستمرة لا تحتاج إلى عملية تقطيع (discretization) كما في خوارزمية REULS-٣ PLUS .

٤- دقة القواعد أكبر من دقتها في REULS-٣ PLUS .

**٥-٣-٤-٢ نقاط الضعف في خوارزمية RULES-٥:**

- يتم اختيار SE بشكل عشوائي (أول نموذج غير معلم ) وذلك دون دراسة للمعطيات.

- للحصول على قواعد مفيدة يجب تطبيق خوارزميات تقليم pruning algorithms بعد تطبيق خوارزمية RULES-٥ وذلك في حال وجود بيانات مشوهة (noisy data) .

- حساب البعد بين SE و CE يتطلب عمليات حساب كبيرة وهذا يتطلب زمناً كبيراً في حال كان عدد السجلات كبيراً .

**٤-٥ خوارزمية شجرة التصنيف والارتداد/Classification And Regression Trees:/CART**

هذه الخوارزمية هي خوارزمية ثنائية (أي تقوم بإنتاج لكل عقدة ابنين اثنين فقط) حيث يتم اختيار مقياس فصل أولي لتحديد العقدة المرشحة Q(s\t).

$$\phi(s\backslash t)=2P_L P_R \sum_{J=1}^{\#class} |P(j\backslash t_L)-P(j\backslash t_R)|$$

حيث إنّ:

$$t_L = left \quad child \quad node \quad of \quad node \quad t$$

$$t_R = right \quad child \quad node \quad of \quad node \quad t$$

$$P_L = \frac{number.of.records.at.t_L}{number.of.records.in.training.set}$$

$$P_R = \frac{number \quad of \quad records \quad at \quad t_R}{number \quad of \quad records \quad in \quad training \quad set}$$

$$P(j\backslash t_L) = \frac{number \quad of \quad class \quad j \quad records \quad at \quad t_L}{number \quad of \quad records \quad at \quad t}$$

$$P(j\backslash t_R) = \frac{number \quad of \quad class \quad j \quad records \quad at \quad t_R}{number \quad of \quad records \quad at \quad t}$$

**جدول (٤،٥)**

| الرمز | الشرح |
|---|---|
| $t_L$ | العقدة الأبن اليسارية للعقدة t |
| $t_R$ | العقدة الأبن اليمينية للعقدة t |
| $p_L$ | نسبة عدد السجلات العقدة اليسارية إلى عدد سجلات التدريب |
| $p_R$ | نسبة عدد السجلات العقدة اليمينية إلى عدد سجلات التدريب |
| $P(j\backslash tL)$ | نسبة عدد سجلات الصنف j في العقدة اليسارية إلى عدد سجلاتt |
| $P(j\backslash t_R)$ | نسبة عدد سجلات الصنف j في العقدة اليمينية إلى عدد سجلاتt |

بعد القيام بحساب العلاقة السابقة يتم اختيار القيمة الأكبر ومن ثم بناء الشجرة حيث يتم إدخال كل القيم في عقدة الجذر ومن ثم يتم تحديد ابنين لها /أيمن وأيسر/ وتستمر عملية الحساب عند كل عقدة حتى الوصـــول إلى صنف ليس بعقدة.

**٥-٥ خوارزمية Decision Tree C٤,٥ Algorithm :**

هذه الخوارزمية هي امتداد لخوارزمية ID٣[٨،١٨] وتستخدم لتوليدDecision Tree، حيث تقوم هـــذه الخوارزمية بزيارة كل عقدة قرار وتحديد القيم التي ستفصل عندها حتى الوصول إلى صنف كمـــا يوضـــح الملحق C آلية عمل الخوارزمية.

ويوجد هناك خلاف بين خوارزميتي C٤,٥ & [٢]CART:

١- الخوارزمية C٤,٥ ليست مخصصة أو منحصرة على التفريق الثنائي بينما CART تنـــتج فقـــط الأشجار الثنائية على عكس خوارزمية C٤,٥ التي تنتج أشجاراً أكثر تعقيداً.

٢- من أجل صفات التصنيف C٤,٥ تنتج افتراضياً أفرعاً منفصلة من أجل كل قيمة، حتى وإن كان بعض القيم تكرارها قليل.

٣- تستخدم خوارزمية C٤,٥ طريقة مختلفة عن خوارزمية CART لتحديد العقدة المـــراد التفريـــق عندها.

حيث تستخدم خوارزمية C٤,٥ ما يسمى بمفهوم ربح المعلومة وذلك مـــن خـــلال تطبيـــق مفهـــوم أنتروبي.

وبفرض لدينا المتغير X والقيم الممكنة K لها احتمالات P١,P٢,.......,Pk وبالتالي يحسب الأنتروبي كما يلي:

$$H(X) = -\sum_{J} Pj * \log(Pj)$$

ومن ثم يتم حساب الأنتروبي للصنف المحتمل أن يكون هو العقدة الفصل فإذا كـــان لـــدينا الفـــصل المرشح هو S بحيث يقسم مجموعة المعطيات T إلى عدة مجموعات للمعطيـــات الفرعيـــة T١,T٢,.....,Tk وبفرض أن Pi تمثل احتمالية من السجلات في المجموعة الفرعية i فيكون حساب الإنتروبي كما يلي:

$$H(X) = -\sum_{i=1}^{k} Pi * Hs(Ti)$$

وبالتالي يتم حساب الربح وفق العلاقة التالية:

$$\text{Gain(S)} = \text{H(T)} - \text{Hs(T)}$$

وتكرر العملية من أجل كل صنف حتى يتم تحديد صاحب الربح الأكبر. يمكن النظر إلى عملية توليد شجرة قرار أولية (ابتدائية) (initial tree) من مجموعة مستندات التدريب على أنها أهم جزء فـــي الخوارزميـــة، و بالنتيجة تنشئ الخوارزمية مصنِّفاً (نموذج تصنيف) على شكل شجرة قرار.

تتكون بنية شجرة القرار المولدة من نوعين من العقد:

١. عقدة ورقة، تعبِّر عن صف التصنيف.

٢. عقدة قرار، تحدد اختباراً ما على قيمة خاصية مفردة مع فرع وحيد و شجرة جزئية لكل ناتج محتمل للاختبار.

و من ثم يمكن استعمال الشجرة الناتجة لتصنيف مستند جديد بدءاً بعقدة الجذر مروراً بالمسارات و وصـــولاً إلى عقدة ورقة تمثل صفاً ينتمي إليه هذا المستند.

تعتمد خوارزمية C٤,٥ على طريقة CLS Hunt's لبناء شجرة قرار من مجموعة مستندات تدريب T. و لكن لنفترض أن الصفوف المحددة مسبقاً هي {C١, ..., Ck} عندها يكون لدينا ثلاث حالات لمحتوى المجموعـــة T:

١. T تحتوي مستنداً أو أكثر جميعها ينتمي إلى الصف Cj. و شجرة القرار هنا هي ورقة تمثل صفاً وحيداً Cj.

٢. T لا تحتوي مستندات، و شجرة القرار هنا هي أيضاً ورقة و لكن لتحديد الصف الذي تمثله هذه الورقة نحتاج إلى معلومات جديدة تزودنا بها مجموعة أخرى غير T. (بمعنى أن T لا تصلح أساساً لتكون مجموعة تدريب)

٣. T تحتوي مستندات تنتمي إلى مجموعة مختلطة من الصفوف. في هذه الحالة الفكرة هي "تنقية" أو تجزيء المجموعة T إلى مجموعات جزئية مكونة من مستندات محاولين الاقتراب من تجمع من المستندات ينتمي إلى صف وحيد و محدد قدر الإمكان.

بالاعتماد على خاصية مفردة، يتم انتقاء اختبار مناسب بحيث تكون نواتجه {O١, ..., On} مبعدة بالتبادل. تُقسَّم T إلى T١, ..., Tn، حيث Ti تتضمن جميع المستندات في T التي لها ناتج Oi للاختبار المعني.

تُطبق هذه الإجرائية على جميع عقد القرار وصولاً إلى عقد الصفوف بطريقة تعاودي.

لا يمكن تحديد عملية بناء الشجرة بشكل فريد. فمن أجل اختبارات مختلفة تُولِّدُ أشجاراً مختلفةً. عادةً نحاول انتقاء الاختبار في مرحلة من التقسيم لتكون الشجرة النهائية صغيرة أو مضغوطة.

لسوء الحظ، مسألة إيجاد أصغر شجرة قرار متناسبة مع مجموعة مستندات التدريب هي عملية معقدة جداً. على سبيل المثال، إذا كان لدينا قاعدة بيانات نصية صغيرة لها خمس خواص و عشرين مستند تدريب فقط، فإن عدد أشجار القرار المحتملة أكبر من ١٠ ملايين شجرة و ذلك استناداً إلى عدد القيم المختلفة لكل خاصية.

لهذا السبب، معظم خوارزميات بناء أشجار القرار غير تراجعية (non - backtracking)، و جشعة (greedy).

لنفترض الآن مهمة إيجاد اختبار محتمل بـ n ناتجاً (n قيمة لخاصية معينة)، يُقسِّم المجموعة T من مستندات التدريب إلى مجموعات جزئية T١, ..., Tn.

المعلومات الوحيدة المتوفرة هي دِلالة على توزيع الصفوف في المجموعة T و مجموعاتها الجزئية Ti.

إذا كانت S أية مجموعة من المستندات، لتكن freq(Ci, S) تعبِّر عن عدد المستندات في S التي تنتمي إلى الصف Ci (من أصل K صفاً محتملاً)، و لتكن |S| تعبر عن عدد المستندات في S.

تَستخدم ID٣ معياراً يسمى العائد (gain) لانتقاء خاصية يُبنى عليها الاختبار تعتمد على مفهوم خاص بنظرية المعلومات (information theory) وهو معدل الإنتروبي (entropy).

تُعطى العلاقة التي تحسب الإنتروبي لمجموعة S كما يلي:

$$\inf o(s)_{1 \le i \le K} = -\sum\left(\left(\frac{freq(c, s_i)}{|s|}\right) * \log_2\left(\frac{freq(c, s_i)}{|s|}\right)\right)$$

اعتبر الآن مقياساً مشابهاً بعد أن قُسمت T آخذين بالاعتبار الـ n ناتجاً لاختبار x وحيد خاص بخاصية معينة.

يمكن الحصول على المتطلب المعلوماتي المتوقع على أنه المجموع الموزون للإنتروبي لكل المجموعات الجزئية:

تقيس المعلومات التي حصلنا عليها كعائد من جراء تقسيم T باعتبار الاختبار x.

معيار العائد يختار الاختبار x لتعظيم Gain(X)، أي يختار الخاصية التي لها أعلى عائد معلومات.

<u>**مثال توضيحي:**</u> لتكن لدينا مجموعة تدريب T بسيطة مكونة من ١٤ عينة (مستند) أو حالة (case)، موصوفة من خلال ثلاث خواص دخل و تنتمي إلى أحد صفين Calss١ أو Class٢. الجدول (٥،٥) يبين هذه المجموعة.

**الجدول (٥،٥) قاعدة بيانات تحتوي بيانات تدريب**

| Attribute١ | Attribute٢ | Attribute٣ | Class |
|---|---|---|---|
| A | ٧٠ | True | Class١ |
| A | ٩٠ | True | Class٢ |
| A | ٨٥ | False | Class٢ |

| | | | |
|---|---|---|---|
| A | ٩٥ | False | Class٢ |
| A | ٧٠ | False | Class١ |
| B | ٩٠ | True | Class١ |
| B | ٧٨ | False | Class١ |
| B | ٦٥ | True | Class١ |
| B | ٧٥ | False | Class١ |
| C | ٨٠ | True | Class٢ |
| C | ٧٠ | True | Class٢ |
| C | ٨٠ | False | Class١ |
| C | ٨٠ | False | Class١ |
| C | ٩٦ | False | Class١ |

نلاحظ أن تسعة مستندات تنتمي إلى الصف الأول و بقية المستندات تنتمي إلى الثاني، و عليه فإن الإنتروبي قبل التقسيم يكون:

$$Info(T) = - ٩/١٤ \log_{٢}(٩/١٤) – ٥/١٤ \log_{٢}(٥/١٤) = ٠,٩٤٠ \text{ bits.}$$

باستعمال الخاصية الأولى لتفريق المجموعة T إلى ثلاث مجموعات جزئية، حيث الاختبار x١ يمثل الاختيار بين ثلاث قيم A, B, or C. نجد أن المعلومات الناتجة هي:

$$Info_{x١}(T) = ٥/١٤ (- ٢/٥ \log_{٢}(٢/٥) – ٣/٥ \log_{٢}(٣/٥))$$
$$+ ٤/١٤ (- ٤/٤ \log_{٢}(٤/٤) – ٠/٤ \log_{٢}(٠/٤))$$
$$+ ٥/١٤ (- ٣/٥ \log_{٢}(٣/٥) – ٢/٥ \log_{٢}(٢/٥)) = ٠,٦٩٤ \text{ bits}$$

و يكون عائد المعلومات من أجل الاختبار x١:

$$Gain(x١) = ٠,٩٤٠ – ٠,٩٦٤ = ٠,٢٤٦ \text{ bits.}$$

إذا كان الاختبار مبنياً على الخاصية الثالثة، حيث x٢ يمثل اختيار واحدة من قيمتين True or False.

$$Infox٢(T) = ٦/١٤ (- ٣/٦ \log٢(٣/٦) – ٣/٦ \log٢(٣/٦))$$

$$+ ٨/١٤ (- ٦/٨ \log٢(٦/٨) – ٢/٨ \log٢(٢/٨)) = ٠,٨٩٢ \text{ bits.}$$

و يكون عائد المعلومات:

$$Gain(x٢) = ٠,٩٤٠ – ٠,٨٩٢ = ٠,٠٤٨ \text{ bits.}$$

بالمقارنة بين معدلي العائدَين للاختبارين x١ و x٢ فإن الخوارزمية ستختار الاختبار الأول ليكون اختبار البدء في تقسيم T و بناء الشجرة لأن عائده أكبر.

من ناحية أخرى، و حتى نتمكن من التعامل مع الاختبار للخاصية الثانية و التي هي خاصية عددية (numeric feature) بقيم مستمرة، يجب أن نوسع مفهوم انتقاء الاختبار.

بشكل عام، تقدم خوارزمية C٤,٥ تقنيات للتعامل مع ثلاثة أنواع من الاختبارات:

١. اختبار قياسي على خاصية لها قيم متقطعة، و ناتج وحيد لفرع وحيد من أجل كل قيمة محتملة (x١ و x٢ في المثال السابق لكل من الخاصية الأولى و الثالثة على الترتيب).

٢. إذا كانت Y خاصية لها قيم عددية مستمرة، يُعتمد اختبار ثنائي بناتجين

$$Y \leq Z \text{ and } Y > Z$$

عن طريق مقارنة قيمتها بقيمة عتبة Z.

٣. اختبار معقد مبني على قيمة متقطعة، بحيث تُسقط القيم المحتملة على عدد متغير من المجموعات، لكل منها ناتج و فرع.

٤. اختبار على القيم المفقودة.

طبعاً ما يهم من أجل الخاصية الثانية هو الحالة الثانية. تتوافر طرق تسهل عملية حساب قيمة مثالية للعتبة Z المطلوبة.

يتم أولاً فرز مستندات التدريب اعتماداً على قيم الخاصية Y. بالتأكيد هناك عدد محدد من هذه القيم.

لتكن {v١, ..., vm} هي تلك المجموعة المرتبة، أي قيمة عتبة تقع بين vi و vi+١ سيكون لها نفس تأثير تقسيم المستندات اعتماداً على قيمة الخاصية Y التي تقع ضمن {v١, ..., vi} و تلك التي تقع ضمن {vi+١, ..., vm}، و بالتالي هناك فقط m-١ جزءاً محتملاً على Y.

عادةً ما يتم اختيار قيمة العتبة لتكون نقطة المنتصف للفترة ٢ / (vi + vi+١). و لكن C٤,٥ تختار تلك القيمة لتكون vi لاعتبارات خاصة بمجال الفترة.

نطبق هذه الخطوات على المثال:

بعد عملية الفرز تكون مجموعة القيم للخاصية الثانية هي {٦٥, ٧٠, ٧٥, ٧٨, ٨٠, ٨٥, ٩٠, ٩٥, ٩٦}.

و مجموعة قيم العتبة المثالية Z (اعتماداً على طريقة C٤,٥) هي {٦٥, ٧٠, ٧٥, ٧٨, ٨٠, ٨٥, ٩٠, ٩٥}.

هنا يجب اختيار قيمة للعتبة Z تكون مثالية من هذه المجموعة، و من أجل المثال تكون Z = ٨٠ و الاختبار x٣ هو Attribute٢ ≤ ٨٠ or Attribute٢ > ٨٠.

و عليه:

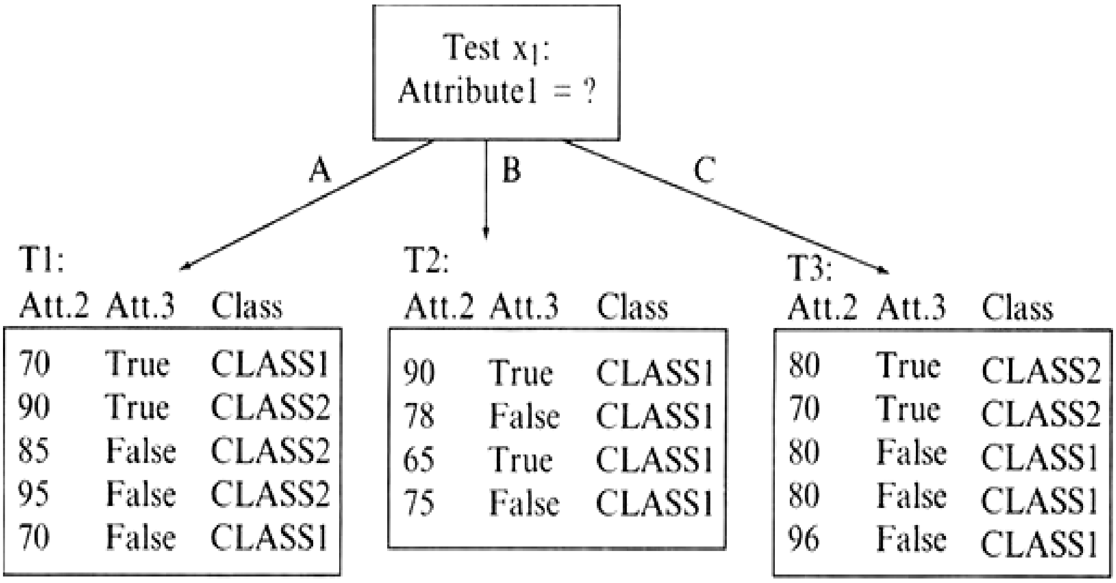
$$Info_{x٣}(T) = ٢/١٤ (- ٧/٩ \log_٢(٧/٩) – ٢/٩ \log_٢(٢/٩))$$

$$+ ٥/١٤ (- ٢/٥ \log_٢(٢/٥) – ٣/٥ \log_٢(٣/٥)) = ٠,٨٣٧ \text{ bits.}$$

وبالتالي يكون الربح :

$$Gain(x٣) = ٠,٩٤٠ – ٠,٨٣٧ = ٠,١٠٣ \text{ bits.}$$

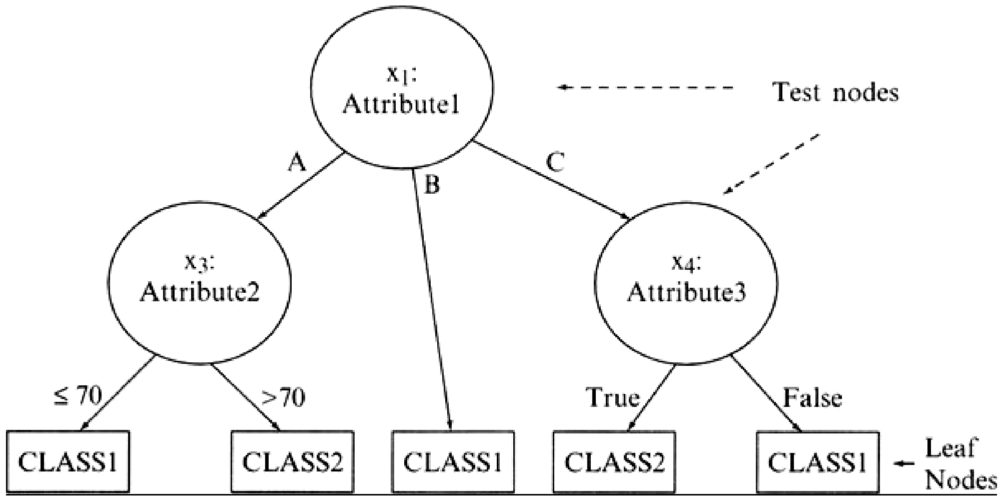
إذا ما قارنّا عائد المعلومات للخواص الثلاثة، سنجد أن الخاصية الأولى ما تزال تقدم أكبر عائد بقيمة ٠,٢٤٦ بت. و بالتالي سيتم اعتمادها لتكون خاصية البداية لإنشاء شجرة القرار. كما في الشكل (٦،٥).




الشكل (٦,٥) شجرة القرار الناتجة

و بما أن العقدة الابن الناتجة عن الاختبار x١ حيث Attribute١ = B لها أربع حـــالات (مـــستندات) تنتمـــي جميعها إلى الصف الأول، فإن هذه العقدة ستكون ورقة و يمكن استثناؤها من باقي الاختبارات.

وبتطبيق نفس الخطوات على الخاصيتين الباقيتين بأسلوب عودي تنتج شجرة قرار نهائية تتضح في الـــشكل التالي:




الشكل (٧،٥) شجرة القرار النهائية

يمكن تمثيل شجرة القرار بطريقة مختلفة، أي باستخدام كود تنفيذي باستخدام قواعد If – then الشرطية لتفريع بنية الشجرة. كما في الشكل (٨،٥).

```
If        Attribute١=A    Then
     If          Attribute٢<=٧٠   Then   Classification=Class١;
                                  Else   Classification=Class٢;
Elseif   Attribute١=B      Then          Classification=Class١;
Elseif   Attribute١=C      Then
     If   Attribute٣=  True        Then   Classification=Class٢;
                                  Else   Classification=Class١;
```

الشكل (٨،٥) شجرة قرار على شكل قواعد

**٥-٦ قيم الخصائص غير المعروفة UNKNOWN ATTRIBUTE VALUES:**

سنركز على التعديلات اللازمة على خوارزمية C٤,٥ لتأخذ بعين الاعتبار قيم الخصائص التي يمكن أن تكون مفقودة لسبب أو لآخر.ولحل هذه المشكلة نحن أمام خيارين :

١-تجاهل الأمثلة في قاعدة المعطيات التي تحوي قيماً مفقودة.

٢-تعريف أو تعديل خوارزمية جديدة لتتعامل مع القيم المفقودة.

ولكن سنستخدم الحل الثاني و عندها ستظهر لدينا بعض المشاكل :

١-كيف ستتم مقارنة مثالين لهما قيم غير معروفة ؟

٢-كيف سيتم تقسيم الأمثلة أثناء بناء الشجرة ؟

لأن الأمثلة التدريبية التي تحوي قيماً غير معروفة لا يمكن أن تترافق مع قيم خاصة للفحص.

في طور الفحص ،كيف سيتم التعامل مع قيمة مفقودة واحدة إذا كان الفحص على الخاصة مع قيم مفقودة ؟

إن خوارزمية C٤,٥ عدلت عملية التقسيم اعتماداً على التكرار للقيم غير المعروفة،حيث يتم حساب Info(T)&Infox(T) كما بالسابق ولكن بدون الأخذ بعين الاعتبار القيم المفقودة ومن ثم يتم حساب عامل الربح و تصحيحه باستخدام العامل F :

Gain(X)=F(Info(T)-Infox(T)) حيث F يمثل نسبة عدد القيم المفقودة إلى القيم الكلية في قاعدة المعطيات .وسنقوم بتطبيق مثال صغير لتوضيح الخوارزمية السابقة :

**الجدول (٦،٥)**

| ATTRIBUTE! | ATTRIBUTE٢ | ATTRIBUTE٣ | CLASS |
|---|---|---|---|
| A | ٧٠ | TRUE | CLASS١ |
| A | ٩٠ | TRUE | CLASS٢ |
| A | ٨٥ | FALSE | CLASS٢ |
| A | ٩٥ | FALSE | CLASS٢ |
| A | ٧٠ | FALSE | CLASS١ |
| ? | ٩٠ | TRUE | CLASS١ |
| B | ٧٨ | FALSE | CLASS١ |
| B | ٦٥ | TRUE | CLASS١ |
| B | ٧٥ | FALSE | CLASS١ |
| C | ٨٠ | TRUE | CLASS٢ |
| C | ٧٠ | TRUE | CLASS٢ |
| C | ٨٠ | FALSE | CLASS١ |
| C | ٨٠ | FALSE | CLASS١ |
| C | ٩٦ | FALSE | CLASS١ |

حساب الربح يتم كما سبق وتم شرحه دون تغيير غير أن القيمة المفقودة تصحح بعض الخطوات السابقة. من الملاحظ أن ٨ أمثلة تنتمي للصنف الأول من أصل ١٣ بينما ينتمي ٥ أمثلة للصنف الثاني فيكون حساب الأنتروبي كما يلي :

$$Info(T) = -8/13\log_2(8/13) - 5/13\log_2(5/13) = 0.961 bits$$

الآن لنحسب الأنتروبي للخاصية X١ :

Info x١(T)=٥/١٣*(−٢/٥*log٢(٢/٥)−٣/٥*log٢(٣/٥))+٣/١٣*(−٣/٣*log٢(٣/٣)- ٠/٣*log٢(٠/٣))

+ ٥/١٣*(−٣/٥*log٢(٣/٥)−٢/٥*log٢(٢/٥))=٠,٧٤٧ bits

فيكون الربح الناتج هو:

Gain(x١)=١٣/١٤(٠,٩٦١−٠,٧٤٧)=٠,١٩٩ bits.

| T١: Attribute=A | | | | T٢: Attribute=B | | | | T٣: Attribute=C | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Att٢ | Att٣ | Class | W | Att٢ | Att٣ | Class | W | Att٢ | Att٣ | Class | W |
| ٧٠ | True | Class١ | ١ | ٩٠ | True | Class١ | ٣/١٣ | ٨٠ | True | Class٢ | ١ |
| ٩٠ | True | Class٢ | ١ | ٧٨ | False | Class١ | ١ | ٧٠ | True | Class٢ | ١ |
| ٨٥ | False | Class٢ | ١ | ٦٥ | True | Class١ | ١ | ٨٠ | False | Class١ | ١ |
| ٩٥ | False | Class٢ | ١ | ٧٥ | False | Class١ | ١ | ٨٠ | False | Class١ | ١ |
| ٧٠ | False | Class١ | ١ | | | | | ٩٦ | False | Class١ | ١ |
| ٩٠ | True | Class١ | ٥/١٣ | | | | | ٩٠ | True | Class١ | ٥/١٣ |

الشكل (٩،٥)

If Attribute١=A Then

If Attribute٢<=٧٠ Then classification=class١ (٢,٠/٠);

Else

classification=class٢ (٣,٤،٠,٤);

ELSEIF Attribute١=B Then classification=class١ (٣,٢/٠);

ELSEIF Attribute١=C Then

If Attribute٣=True Then classification=class٢ (٢,٤/٠,٤);

Else classification=class١ (٣,٠/٠);

ولكن بسبب الغموض الناتج في القوانين الأخيرة نجد أن القرار له بارا مترين(Ti/E) حيث Ti تمثل مجموع الأمثلة الجزئية التي وصلت للورقة بينما E تمثل عدد الأمثلة التي تنتمي للأصناف غير الصنف المسمى والمدروس.

على سبيل المثال (٣,٤/٠,٤) تعني أن ٣,٤ أو (٣+٥/١٣) المثال المدرب الذي يصل للورقة حيث ٠,٤(٥/١٣) لا تنتمي للصنف الذي يحدد الورقة.

٨٨٪ = ١٠٠٪٠ ٣/٣,٤ احتمال أن الورقة الناتجة تنتمي للصنف الثاني CLASS٢.

١٢٪ = ١٠٠٪٠ ٠,٤/٣,٤ احتمال أن الورقة تصنف الصنف الأول CLASS١ .

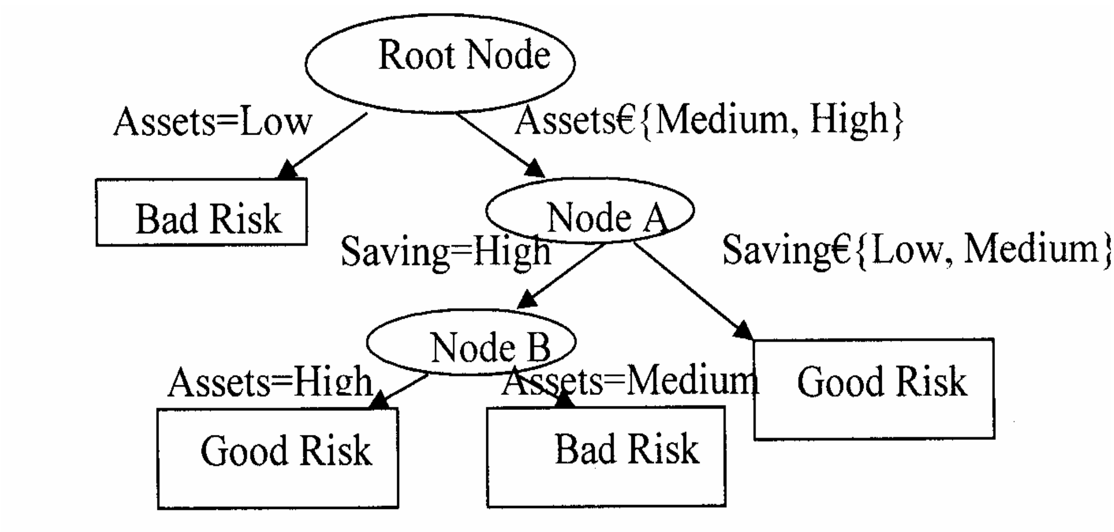
**٥-٧ تطبيق كل من خوارزمية CART وخوارزمية C٤,٥ والمقارنة بينهما:**

سنبين فيما يلي المثال التالي ونحدد الشجرة الناتجة لكل من الخوارزميتين:

**الجدول رقم (٥،٧)**

| **Saving** | **Assets** | **Income($١٠٠٠S)** | **Credit Risk** |
|---|---|---|---|
| Medium | High | ٧٥ | Good |
| Low | Low | ٥٠ | Bad |
| High | Medium | ٢٥ | Bad |
| Medium | Medium | ٥٠ | Good |
| Low | Medium | ١٠٠ | Good |
| High | High | ٢٥ | Good |
| Low | Low | ٢٥ | Bad |
| High | Medium | ٧٥ | Good |

بتطبيق المثال السابق على خوارزمية CART تنتج الشجرة التالية:




الشكل (١١،٥)

وبالتالي فإن القوانين الناتجة من الشجرة السابقة ستكون كما يلي:

If Assets =Low Then Bad Risk

Else If (Assets=Medium OR Assets =High) Then

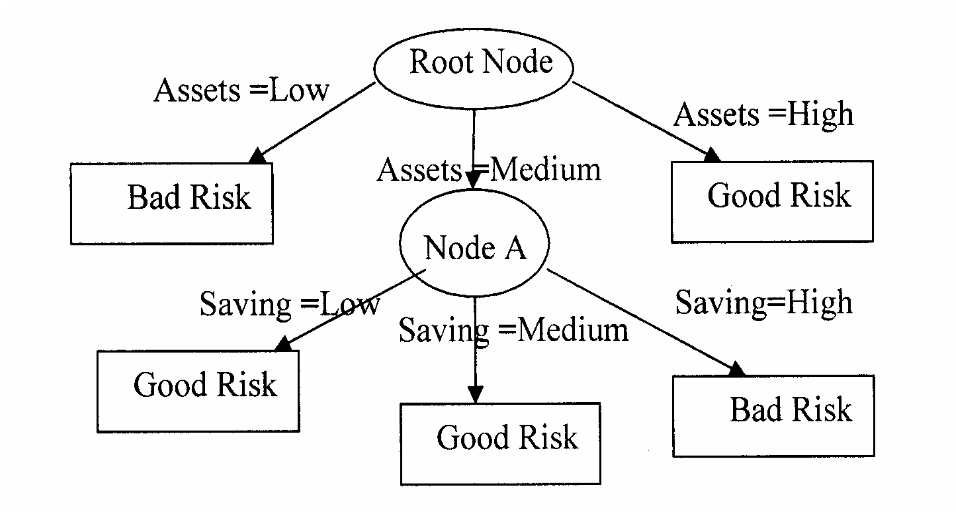
If (Saving =Low OR Saving=High) Then Good Risk

Else If Saving =High Then

If Assets=High Then Good Risk

Else If Assets =Medium Then Bad Risk

وبتطبيق نفس المثال السابق على خوارزمية C٤,٥ تنتج الشجرة التالية:




الشكل رقم (١٢،٥)

وبالتالي ستكون القوانين المولدة من الشجرة السابقة كما يلي:

If Assets =Low Then Bad Risk
Else If Assets =High Then Good Risk
Else If =Medium Then
If Saving=Low Then Good Risk
Else If Saving=Medium Then Good Risk
Else If Saving = High Then Bad Risk

نلاحظ من النتائج الحاصلة من الشجرتين بأن طريقة CART تقوم بتوليد شجرة ثنائية فقط وتنتج قوانين كثيرة بالإضافة إلى أنها يمكن أن تكرر فحص نفس الصفات في مستويات مختلفة بينما طريقة C٤,٥ هي تقوم برسم الشجرة بعدة فروع كما أنها أسرع وتغطي بقوانينها القليلة كامل المثال ولا يمكن إعادة فحص صفة في مستوى معين إذا تم فحصه في مستوى سابق.

### ٥-٨ تشذيب قوانين شجرة التصنيف Pruning Decision Tree Rules:

رغم سرعة التصنيف وتوليد القوانين لدى خوارزمية شجرة التصنيف C٤,٥ ورغم شموليتها إلا أن قوانينها تحتاج لتشذيب ويمكن أن يتم التشذيب [٢٣-٢١] في إحدى حالتين:

- ✓ تشذيب القوانين قبل عملية التصنيف Pre-pruning.
- ✓ تشذيب القوانين بعد عملية التصنيف Post-pruning.

ويوجد الكثير من الخوارزميات التي تقوم بعملية تشذيب قوانين التصنيف ولذلك تم هناك الكثير من المحاولات لتوليد خوارزميات للقيام بمهمة تشذيب القوانين.

إن خوارزمية Rule٣ تقوم باستنباط القوانين والمعرفة وهي تقوم بتوليد قوانين مشذبة نوعاً ما وشاملة ولكن هذه الخوارزمية بطيئة جداً في توليد القوانين وهي تقوم بتشذيب قوانين شجرة التصنيف C٤,٥ وذلك بتمرير القوانين الناتجة من خوارزمية C٤,٥ إلى خوارزمية Rule٣ وبالتالي تحسين وتشذيب قوانين الشجرة.

# الفصل السادس

**Developing of C٤,٥ Algorithm**

١– تحسين فعالية شجرة التصنيف.

٢– تشذيب قوانين شجرة التصنيف.

٣– مقارنة التعديلات الناتجة مع الناتج من تطبيق خرج خوارزمية C٤,٥ على خوارزمية Rule٣.

٤- مقارنة التعديلات الناتجة مع خوارزمية Rule٥.

### ٦-١ تعديل خوارزمية C٤,٥:

#### ٦-١-١التعديل الأول لتحسين فعالية خوارزمية C٤,٥:

من الملاحظ أنه رغم شمولية خوارزمية C٤,٥ فإنها تكلف كثيراً في الزمن في حساب الأنتروبي لكل الصفات عند كل عقدة ولذلك تم اقتراح إضافة تعديل وهو أنه يتم أخذ الصفة التي تحوي قيماً فريدة أكثر لتكون هي العقدة المفحوصة وفي حال وجود أكثر من صفة تملك نفس العدد الفريد من القيم نقوم بحساب الأنتروبي والربح لهذه الصفات ومن ثم اختيار الصفة التي تملك ربحاً أكبر.كما وتم تطبيق تعديل آخر على خوارزمية شجرة التصنيف يؤدي إلى توليد قوانين مشذبة والوصول إلى نتائج مشذبة وبسرعة.

إن التعديل المقترح يقوم على البحث ضمن كل صفة عن القيم التي تنتمي إلى صنف واحد فقط وإذا كان عدد القيم أكثر من واحد يتم إضافة هذه القيم كقوانين ومن ثم يتم حذف كامل الأمثلة التي تحقق القوانين الناتجة السابقة من جدول المعطيات وبعد ذلك يتم تمرير المعطيات المتبقية إلى خوارزمية C٤,٥ لإنتاج بقية القوانين، كما يوضح الملحق C مخططاً بآلية عمل خوارزمية C٤,٥ وتعديلها.

وسنقوم الآن بتطبيق خوارزمية C٤,٥ وتعديلها المقترح على بيانات مثال الطقس/ Witte and Frank ١٩٩٩/ التي تحدد إمكانية اللعب أم لا وفقاً لحالة الطقس والمبينة في الجدول التالي:

جدول (٦-١)

| Outlook | Temperature | Humidity | Windy | Play |
|---|---|---|---|---|
| Sunny | Hot | High | False | No |
| Sunny | Hot | High | True | NO |
| Overcast | Hot | High | False | Yes |
| Rainy | Mild | High | False | Yes |
| Rainy | Cool | Normal | False | Yes |
| Rainy | Cool | Normal | True | No |
| Overcast | Cool | Normal | True | Yes |
| Sunny | Mild | High | False | No |
| Sunny | Cool | Normal | False | Yes |
| Rainy | Mild | Normal | False | Yes |
| Sunny | Mild | Normal | True | Yes |

| | | | | |
|---|---|---|---|---|
| Overcast | Mild | High | True | Yes |
| Overcast | Hot | Normal | False | Yes |
| Rainy | Mild | High | True | No |

سنقوم من خلال تطبيق المثال السابق على خوارزمية C٤,٥ بحساب الأنتروبي للصنف Play ومن ثم حساب الأنتروبي من أجل كل الصفات ومن ثم حساب الربح لتحديد عقدة الجذر وبالحساب نجد أن الــصفة الأكثر ربحاً هي صفة Outlook ونقوم بتقسيم وتفريع القيم على أساس قيم الصفة السابقة ثم تكرار الحــساب عند كل عقدة ناتجة وبالنتيجة ستنتج الشجرة التالية بعد المرور بالمراحل والحسابات التالية مع العلم بأن هذه النتائج تم أخذها من خلال تطبيق الجدول رقم /١،٦/ على البرنامج الذي تم عمله باستخدام لغة C# :

H (T)=ROUND(-(٥/١٤*LOG(٥/١٤;٢)+٩/١٤*LOG(٩/١٤;٢));٣)=٠,٩٤.

H(OUTLOOK,T)=ROUND(٥/١٤*(٣/٥*LOG(٣/٥;٢)+٢/٥*LOG(٢/٥;٢))-

٤/١٤ * (٤/٤*LOG(٤/٤;٢)+٠/٤) - ٥/١٤ * (٣/٥*LOG(٣/٥;٢)+٢/٥ *

LOG(٢/٥;٢));٣)=٠,٦٩٤.

H(TEMPERATURE,T)=ROUND(٤/١٤*(٢/٤*LOG(٢/٤;٢)+٢/٤*LOG(٢/٤;٢))-

٦/١٤*(٤/٦*LOG(٤/٦;٢)+٢/٦*LOG(٢/٦;٢))-

٤/١٤*(٣/٤*LOG(٣/٤;٢)+١/٤*LOG(١/٤;٢));٣)=٠,٩١١.

H(HUMIDITY,T)=ROUND(٧/١٤*(٤/٧*LOG(٤/٧;٢)+٣/٧*LOG(٣/٧;٢))-

٧/١٤*(٦/٤*LOG(٦/٧;٢)+١/٧*LOG(١/٧;٢));٣)=٠,٨٦.

H(WINDY,T)=ROUND(٨/١٤*(٢/٨*LOG(٢/٨;٢)+٦/٨*LOG(٦/٨;٢))٦/١٤*(٣/٦*LOG(٣/٦;

٢)+٣/٦*LOG(٣/٦;٢));٣)=٠,٨٩٢

ثم سنقوم بحساب الربح من أجب كل صفة:

GAIN (OUTLOOK,T)=H(T)-H(OUTLOOK,T)=٠,٩٤−٠,٦٩٤=٠,٢٤٦.

GAIN(TEMPERATURE,T)=H(T)-H(TEMPERATURE,T)=٠,٩٤−٠,٩١١=٠,٠٢٩.

GAIN(HUMIDITY,T)=H(T)-H(HUMIDITY,T)=٠,٩٤−٠,٨٦=٠,٠٨.

GAIN(WINDY,T)=H(T)-H(WINDY,T)=٠,٩٤−٠,٨٩٢=٠,٠٤٨.

من هذه الحسابات نجد أن عقدة الجذر المختارة هي عقدة OUTLOOK.

ونلاحظ أيضاً أنه عندما تكون Outlook=overcast سيكون هناك صنف واحد وهو YES.

وعندما تكون OULOOK=Sunny سنقوم بالحسابات التالية لتحديد عقدة الجذر والتفريق الجديدة:

H(T) =ROUND(-٣/٥*LOG(٣/٥;٢)-٢/٥*LOG(٢/٥;٢);٣)=٠,٩٧١.

H(TEMPERATURE,T) =ROUND(-٢/٥*LOG(٢/٢;٢)-٢/٥*(LOG(١/٢;٢)+١/٢*LOG(١/٢;٢))-

١/٥*LOG(١/١;٢);٣)=٠,٦.

H(HUMIDITY,T)=ROUND(٣/٥*(٢/٣*LOG(٢/٣;٢)+١/٣*LOG(١/٣;٢))-

٢/٥*(٢/٢*LOG(٢/٢;٢));٣)=٠,٥٥١.

H(WINDY,T)=ROUND(-٣/٥*(٢/٣*LOG(٢/٣;٢)+١/٣*LOG(١/٣;٢))-

٢/٥*(١/٢*LOG(١/٢;٢)+١/٢*LOG(١/٢;٢));٣)=٠,٩٥١.

GAIN(TEMPERATURE,T)=H(T)-H(TEMPERATURE,T)=٠,٩٧١-٠,٦=٠,٣٧١.

GAIN(HUMIDITY,T)=H(T)-H(HUMIDITY,T)=٠,٩٧١-٠,٥٥١=٠,٤٢.

GAIN(WINDY,T)=H(T)-H(WINDY,T)=٠,٩٧١-٠,٩٥١=٠,٠٢.

من الحسابات السابقة نجد أن عقدة الجذر هي صاحبة الربح الأكبر وهي Humidity، وعند التفريق عندها نجد أنه عندما تكون Humidity= High يكون النتيجة هي NO وعندما تكون Humidity=Normal عندها ستكون النتيجة هي YES.

أما عندما يكون Outlook=Rainy سنقوم بالحسابات التالية:

H(T) =ROUND(-٣/٥*LOG(٣/٥;٢)-٢/٥*LOG(٢/٥;٢);٣)=٠,٩٧١.

H(TEMPERATURE;T)=ROUND(٣/٥*(٢/٣*LOG(٢/٣;٢)+١/٣*LOG(١/٣;٢))-

٢/٥*(١/٢*LOG(١/٢;٢)+١/٢*LOG(١/٢;٢));٣)=٠,٩٥١.

H(HUMIDITY,T)=ROUND(٢/٥*(١/٢*LOG(١/٢;٢)+١/٢*LOG(١/٢;٢))-

٣/٥*(٢/٣*LOG(٢/٣;٢)+١/٣*LOG(١/٣;٢));٣)=٠,٩٥١.

H(WINDY,T) =ROUND(−٣/٥*(٣/٣*LOG(٣/٣;٢))−٢/٥*(٢/٢*LOG(٢/٢;٢));٣)=٠.

GAIN(TEMPERATURE,T)=H(T)-H(TEMPERATURE,T)=٠,٩٧١−٠,٩٥١=٠,٠٢.

GAIN(HUMIDITY,T)=H(T)-H(HUMIDITY,T)=٠,٩٧١−٠,٩٥١=٠,٠٢.

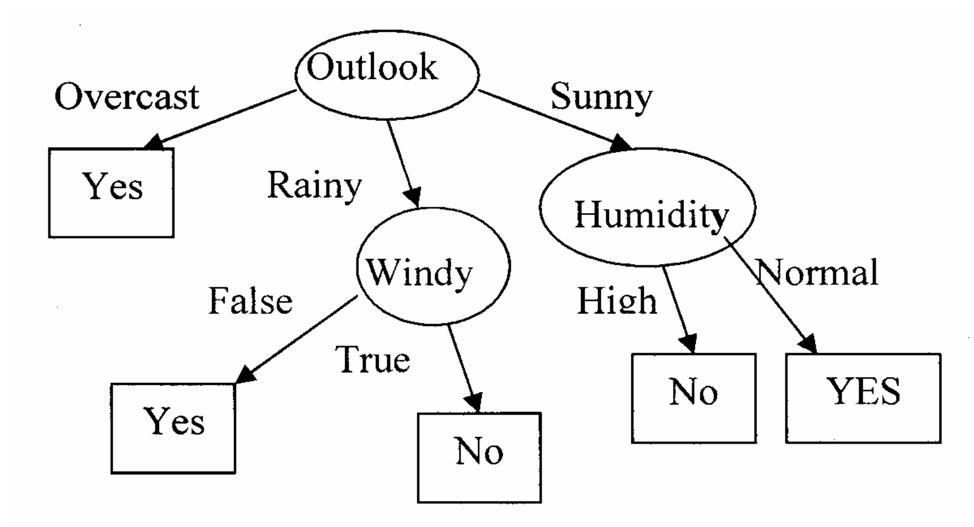
GAIN(WINDY,T)=H(T)-H(WINDY,T)=٠,٩٧١−٠=٠,٩٧١.

ومن هذه الحسابات السابقة نجد أن الصفة ذات الربح الأكبر هي العقدة WINDY حيث عندما تكون قيمتها Windy=True عندها النتيجة هي NO وعندما تكون Windy=False ستكون النتيجة هي YES.

```
|outlook|
          <overcast>
          |play|
          <rainy>
          |windy|
                    <no>
                    |play|
                    <yes>
                    |not play|
          <sunny>
          |humidity|
                    <high>
                    |not play|
                    <normal>
                    |play|
```

الشكل (٦،١)

**نتيجة تطبيق المثال على البرنامج**




الشكل (٦،٢)

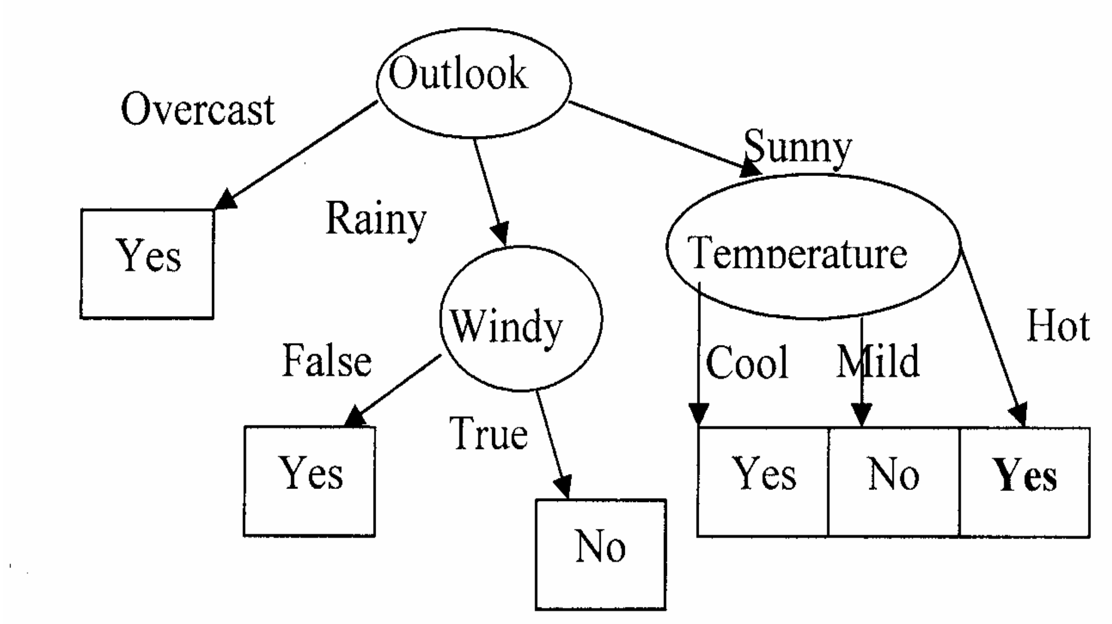
ستكون القوانين الناتجة كما يلي:

```
If Outlook=Overcast Then Play
    Else If Outlook=Rainy Then
        If Windy=False Then Play
        Else If Windy=True Then Not Play
 Else If Outlook=Sunny Then
        If Humidity=High Then Not Play
        Else If Humidity=Normal Then Not Play
```

حيث كما هو ملاحظ من الشجرة المرسومة بأن حساب الأنتروبي تم عند عقدة الجذر Outlook خمس مرات من أجل خمس صفات ثم تم حساب الأنتروبي عند كل من عقدة Windyوعقدة Humidity أربع مرات بسبب وجود أربع صفات كما تم حساب الربح عند كل عملية حساب للأنتروبي. وإن هذه الخوارزمية تخضع لفرع التعلم بإشراف Supervised Learning أي إذا تم بناء الشجرة على أمثلة معينة وتم أثناء عملية الفحص إعطاء حالة غير موجودة في مرحلة التدريب سيتم إعطاء أخطاء كما تتميز هذه الطريقة بأنه كلما كبرت قاعدة المعطيات كلما أصبحت الشجرة أكبر ولا تكرار فيها.

وبتمرير نفس المثال السابق على خوارزمية C٤,٥ مع التعديل المقترح الذي ينص على أنه يم اختيار الصفة التي تحوي قيماً فريدة أكثر كعقدة جذر وإذا ما وجد أكثر من صفة تملك نفس عدد القيم الفريدة يتم حساب الأنتروبي لهم بعد ذلك يتم اختيار الصفة التي تعطي ربحاً أكثر فنكون بذلك قد خفضنا عدد المرات التي سيتم بها حساب الأنتروبي وبالتالي إنقاص الزمن اللازم لذلك.

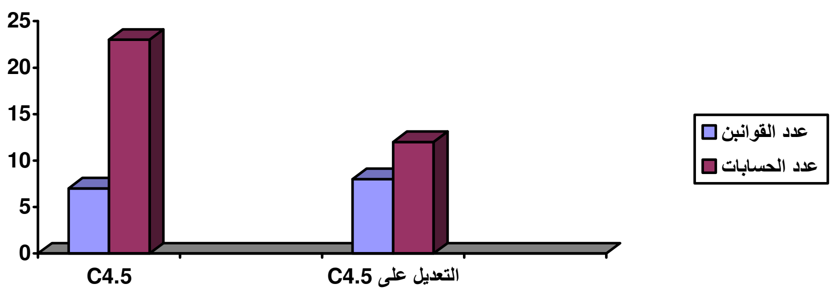
نجد أن بهذا التمرير تم في عقدة الجذر حساب الأنتروبي فقط لصفتي Temperature & Outlook وللصنف ومن ثم في المستوى التالي عندما يكون Outlook=overcast لن يتم حساب الأنتروبي ولا مرة كما أنه في حالة Outlook=sunny نجد أن عدد القيم الفريدة Temperature أكثر من غيرها وبالتالي لم نحسب الأنتروبي، ولكن في الحالة الأخيرة Outlook=rainy عندها سيتم حساب الأنتروبي لجميع الصفات لأنها كلها تحوي نفس عدد القيم الفريدة ومنه ستنتج الشجرة التالية:




الشكل (٣،٦)

وبالتالي ستكون القوانين الناتجة كما يلي:

If Outlook=Overcast Then Play
Else If Outlook=Rainy Then
If Windy=False Then Play
Else If Windy=True Then Not Play
Else If Outlook=Sunny Then
Else If Temperature=Cool Then Play
Else If Temperature=Mild Then Not Play
Else If Temperature=Hot Then Play.



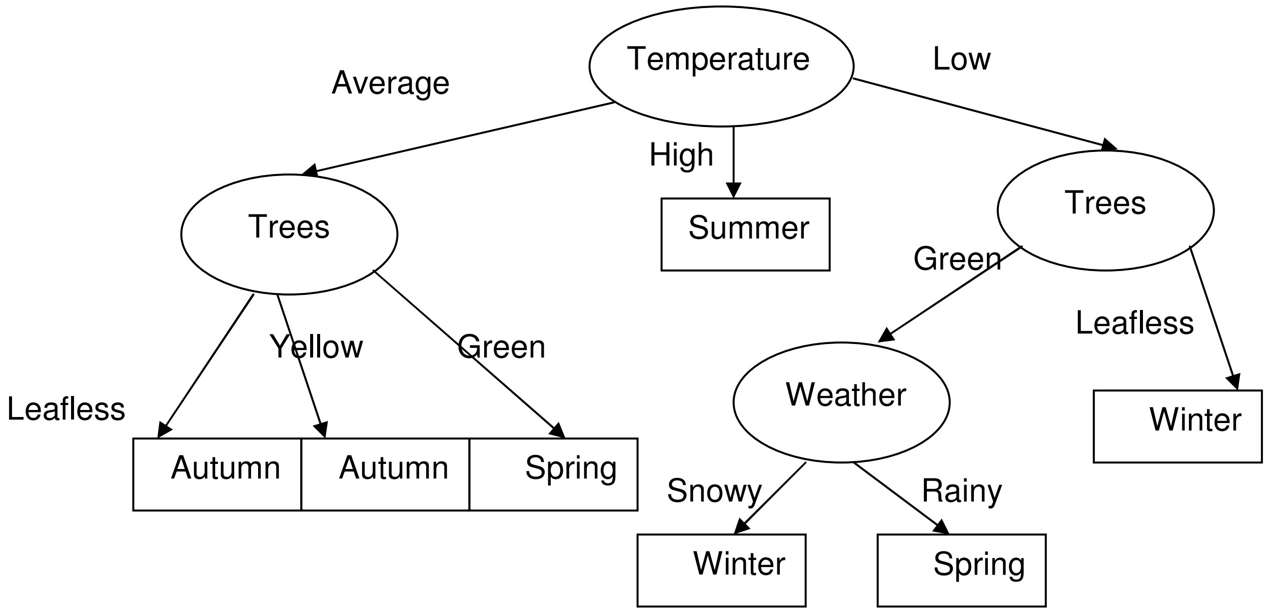

المنحني رقم (١،٦) مقارنة بين الخوارزمية وتعديلها

### ٦-١-١-أ تطبيق الخوارزمية وتعديلها على المثال التالي:

جدول رقم (٢،٦)

| Weather | Tree | Temperature | Season |
|---|---|---|---|
| Rainy | Yellow | Average | Autumn |
| Sunny | Yellow | Average | Autumn |
| Rainy | Leafless | Ave rage | Autumn |
| Rainy | Green | Average | Spring |
| Sunny | Green | Average | Spring |
| Sunny | Leafless | Low | Winter |
| Snowy | Leafless | Low | Winter |
| Rainy | Leafless | Low | Winter |
| Snowy | Green | Low | Winter |
| Rainy | Green | Low | Spring |
| Sunny | Green | High | Summer |
| Rainy | Green | High | Summer |

وبتطبيق بيانات الجدول السابق على خوارزمية C٤,٥ ستنتج الشجرة التالية:




الشكل رقم (٤،٦)

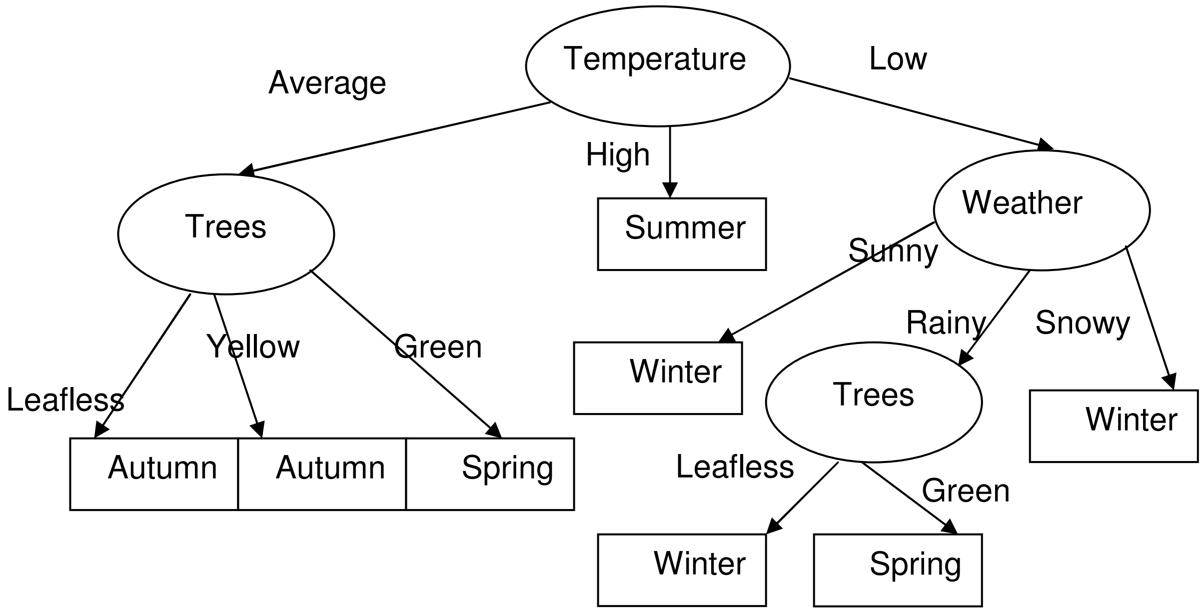
وبالتالي ستكون القوانين كما يلي:

```
If Temperature=High Then Sunny
Else If Temperature=Average Then
    If Trees=Leafless Then Autumn
    Else If Trees=Yellow Then Autumn
    Else If Trees=Green Then Spring
Else If Temperature=Low Then
    If Trees=Leafless Then Winter
    Else If Trees=Green Then
        If Weather=Snowy Then Winter
        Else If Weather=Rainy Then Spring
```

وبتطبيق المثال في الجدول (٢،٦) على التعديل المقترح نجد الشجرة في الشكل (٥،٦).

حيث من الملاحظ أنه عند عقدة الجذر الأول نجد أن الأصناف الثلاثة تملك نفس العدد الفريد من القيم لذلك يتم حساب الأنتروبي للصفات الثلاثة ومن ثم اختيار الصفة Temperature التي تملك الربح الأكبر وفي المرحلة التالية نجد أنه عندما كانت Temperature=high تعطي الصنف Summer مباشرةً وعندما يكون Temperature=average نجد أن الصفة Trees تملك أكبر عدد من الصفات الفريدة لذلك يتم اختياره كعقدة جذر ومنها ينتج صفات مباشرة بينما عندما يكون Temperature=low يتم اختيار عقدة الجذر Weather التي بدورها تعطي صفاً مباشراً إلا عندما تكون قيمتها Rainy فسيتم فحص الصفة Trees.

من المثال السابق نجد أن التعديل أعطى نفس عقدة الجذر في المستوى الأول ونفس العقدة عندما كانت الصفة Temperature=average ولكن تولد خلاف في عقدة الجذر الأخرى حيث تم اختيار الصفة Weather بدلاً من Trees وبعد ذلك تم فحص الصفة Trees عندما دعت الحاجة على عكس خوارزمية C٤,٥.




الشكل (٦،٥)

وبالتالي ستكون القوانين كما يلي:

If Temperature=High Then Sunny

Else If Temperature=Average Then

    If Trees=Leafless Then Autumn

    Else If Trees=Yellow Then Autumn

    Else If Trees=Green Then Spring

Else If Temperature=Low Then

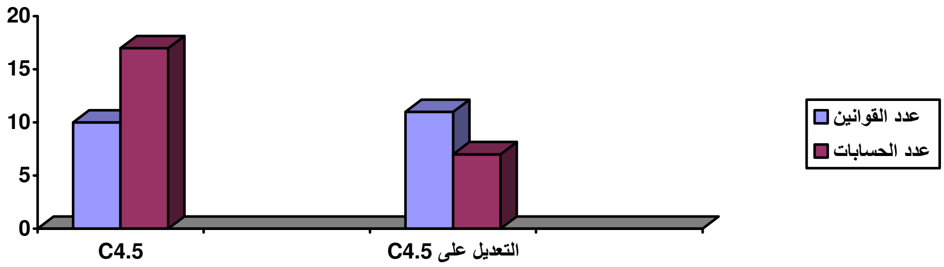
    If Weather=Sunny Then Winter

    Else If Weather=Snowy Then Winter

    Else If Weather=Rainy Then

        If Trees=Green Then Spring

        Else If Trees=Leafless Then Winter




**المنحني رقم (٦،٢) مقارنة بين الخوارزمية C٤,٥ وتعديلها**

**٦-١-٢ التعديل الثاني لتشذيب قوانين خوارزمية C٤,٥:**

هذا التعديل يقترح بأن يتم سحب القيم التي تنتمي لصنف وحيد ضمن كل صفة بعد ذلك اعتبار الأمثلة المسحوبة قوانيناً ثم حذفها من الجدول، حيث يوضح المخطط الموجود في الملحق C آلية العمل.

**٦-١-٢-١ تطبيق الخوارزمية وتعديلها على المثال التالي :**

من النظر للجدول (١،٦) نجد بأن الجدول لا يحوي سوى قيمة outlook=overcast وبالتالي لا داعي لتطبيق التعديل المقترح للتشذيب.

**٦-١-٢-١-أ تطبيق الخوارزمية وتعديلها على المثال التالي :**

من المثال الموضح في الجدول (٢،٦) نجد أن القوانين الناتجة من تطبيقه على خوارزمية C٤,٥ موضحة في الجدول التالي:

**الجدول رقم (٣،٦)**

| Weather | Tress | Temperature | season |
|---|---|---|---|
| * | * | High | Summer |
| * | Leafless | Average | Autumn |
| * | Yellow | Average | Autumn |
| * | green | average | spring |
| * | Leafless | low | Winter |
| snowy | Green | Low | Winter |
| rainy | Green | Low | Spring |

وبتطبيق المثال في الجدول (٢،٦) على التعديل المقترح نجد أن الصفة weather=snowy تحدد صنفاً وحيداً وهو winter كما أن الصنف trees=yellow أيضاً يحدد صنفاً وحيداً وهو autumn كما وأن الصفة temperature=high تغطي صنفاً وحيداً وهو summer لذلك سيتم ضم القوانين الثلاثة المؤلفة كل منها من صفة تعطي صنفاً وحيداً ثم ستحذف كل هذه الأمثلة /٦ أمثلة/ من قاعدة المعطيات المدروسة وتدخل الأمثلة المتبقية إلى خوارزمية C٤,٥ فتنتج الشجرة التالية:

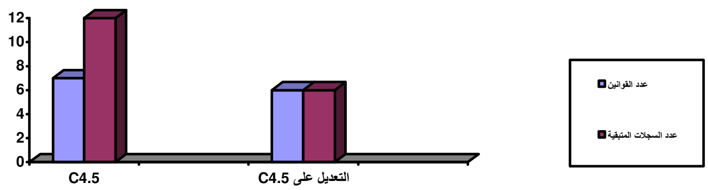
```
|trees|
        <green>
        |spring|
        <leafless>
        |temperature|
                <average>
                |autumn|
                <low>
                |winter|
```

الشكل ( ٦،٦)

وبالتالي ستكون القوانين كما يلي:

**الجدول رقم (٤،٦)**

| Weather | Tress | Temperature | season |
|---|---|---|---|
| snowy | * | * | winter |
| * | yellow | * | Autumn |
| * | * | high | summer |
| * | green | * | spring |
| * | Leafless | low | Winter |
| * | leafless | average | autumn |




**المنحني رقم (٣،٦)**

من المنحني السابق رقم (٣،٦) والجدولين رقم (٣،٦) و (٤،٦) نجد أن القوانين الناتجة عن تعـــديل خوارزمية C٤,٥ أعطى قوانين مشذبة واستبعد القيم غير المناسبة مع ملاحظة إنقاص عـــدد القـــوانين عـــن الخوارزمية الأصلية بمقدار واحد حيث إن أول ثلاثة قوانين من الجدول رقم /٦،٤/ نتجت من التعديل.

### ٦-١-٢-١-ب تطبيق الخوارزمية وتعديلها على المثال التالي :

**الجدول (٥،٦)**

| Example no | CarSize | Fuel | MaxSpeed | Acceleration |
|---|---|---|---|---|
| ١ | Large | Diesel | High | Good |
| ٢ | Large | Propane | High | Good |
| ٣ | Compact | Petrol | High | Excellent |
| ٤ | Large | Petrol | High | Excellent |

| | | | | |
|---|---|---|---|---|
| ٥ | Medium | Diesel | Low | Good |
| ٦ | Compact | Petrol | Low | Good |
| ٧ | Medium | Petrol | Average | Excellent |
| ٨ | Medium | Diesel | Average | Poor |

تنتج الشجرة التالية من نتيجة تطبيق خوارزمية C٤,٥ على المثال السابق:

```
|fuel|
        <diesel>
        |maxspeed|
                <average>
                |poor|
                <high>
                |good|
                <low>
                |good|
        <petrol>
        |maxspeed|
                <average>
                |excellent|
                <high>
                |excellent|
                <low>
                |good|
        <propane>
        |good|
```

الشكل (٧،٦)

وبالتالي ستنتج القوانين التالية:

**الجدول (٦،٦)**

| **Car-size** | **Fuel** | **Max-speed** | **acceleration** |
|---|---|---|---|
| * | Diesel | Average | Poor |
| * | Diesel | High | Good |
| * | Diesel | Low | Good |
| * | Petrol | Average | Excellent |
| * | Petrol | High | Excellent |
| * | Petrol | Low | Good |
| * | Propane | * | Good |

وبتطبيق المثال السابق على التعديل المقترح نجد أن الصفة max-speed=low تحدد صـــنفاً واحــداً وبالتالي سيتم إضافته للقوانين وحذف جميع الأمثلة المحققة له/٢ مثال/ من الجدول /٥،٦/ كما وأن الــصفة

Fuel=Propane تعطي الصنف Good وهو وحيد أيضاً سيضاف للقوانين ويحذف من الجدول/مثـال واحـد/ وبالتالي سندخل الجدول المتبقية/٥ أمثلة/ إلى خوارزمية C٤,٥ لتنتج الشجرة التالية:

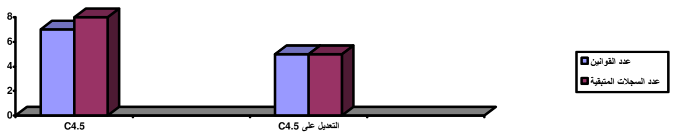
```
|fuel|
        <diesel>
        |maxspeed|
                <average>
                |poor|
                <high>
                |good|
        <petrol>
        |excellent|
```

الشكل (٨،٦)

وبالتالي ستنتج القوانين التالية:

**الجدول (٧،٦)**

| Car-size | Fuel | Max-speed | Acceleration |
|---|---|---|---|
| * | Propane | * | Good |
| * | * | Low | Good |
| * | Petrol | * | Excellent |
| * | Diesel | High | Good |
| * | Diesel | Average | Poor |




**المنحني رقم (٤،٦) مقارنة بين الخوارزمية C٤,٥ وتعديلها**

من المنحني السابق رقم (٤،٦) ومن الجدولين(٦،٦) و(٧،٦) نجد نقصان عـدد القـوانين بمقـدار قانونين مع العلم أن القانونين الأول و الثاني من الجدول /٧/ نتجا من التعديل المقترح وبالتالي تم الحـصول على قوانين أكثر شمولية وتشذيباً باستخدام التعديل المقترح.

### ٦-٦-٢-١-ت تطبيق الخوارزمية وتعديلها على المثال التالي:

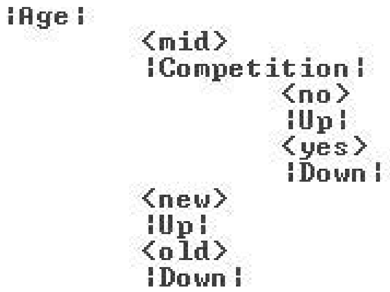
وهذا المثال يمثل Stock Market حيث swr=software و hwr=hardeware

**جدول (٨،٦)**

| Age | Competition | Type | Profit |
|---|---|---|---|
| old | yes | swr | down |
| old | no | swr | down |
| old | no | hwr | down |
| mid | yes | swr | down |
| mid | yes | hwr | down |
| mid | no | hwr | up |

| | | | |
|---|---|---|---|
| mid | no | swr | up |
| New | yes | swr | up |
| new | no | hwr | up |
| new | no | swr | up |

بتطبيق المثال السابق في الجدول (٦،٨) تنتج الشجرة التالية:




الشكل (٦،٩)

وبالتالي تنتج القوانين التالية كما في الجدول (٦،٩):

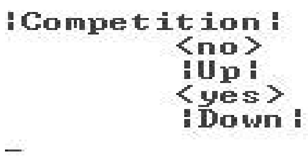
**جدول (٦،٩)**

| Age | Competition | Type | Profit |
|---|---|---|---|
| mid | no | * | up |
| mid | yes | * | down |
| new | * | * | up |
| old | * | * | down |

If Age=mid and Competition then Profit=up
Else if Age=mid and Competition=yes then Profit=down
Else if Age=new then Profit=up
Else if Age=old then Profit=down.

وبعد تطبيق الجدول على التعديل نجد أن الصفة Age تملك القيمتين old و new تعطي صنفاً واحداً أما عندما وبالتالي يمكن اعتبار القيمتين قانوناً وحذف السجلات المحققة لهما ثم ادخال المتبقي من قاعدة البيانات إلـــى خوارزمية C٤,٥.

وبعد تطبيق الجدول رقم /٦ ،٨/ على التعديل المقترح للتشذيب تنتج الشجرة التالية:




الشكل (٦،١٠)

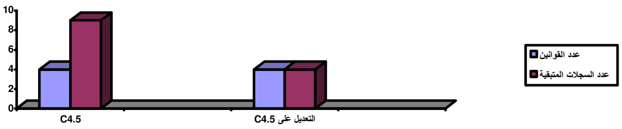
هذا علماً أنه بحسب التعديل سيتم اعتبار قيمة الخاصية Age=old قانوناً وحذف السجلات الثلاثة المحققة للقانون وسيتم اعتبار Age=new قانون وحذف السجلات الثلاثة من المعطيات وبالتالي سيتم تطبيق أربعة سجلات فقط على خوارزمية C٤,٥.

وبالتالي ستكون القوانين الناتجة كمايلي:

جدول (١٠،٦)

| Age | Competition | Type | Profit |
|---|---|---|---|
| * | no | * | up |
| * | yes | * | down |
| new | * | * | up |
| old | * | * | down |

If Competition=no then Profit=up
Else if Competition=yes then Profit=down
Else if Age=new then Profit=up
Else if Age=old then Profit=down.




المنحني رقم (٥،٦) مقارنة بين الخوارزمية C٤,٥ وتعديلها

ومن المقارنة نجد أن القوانين أصبحت أكثر اختصاراً وتشذيباً.

**٦-٢ مقارنة النتائج السابقة الذكر مع النتائج المأخوذة من تطبيق Rule٣ على خرج C٤,٥:**

a) <u>تطبيق Rule٣ على بيانات الجدول (٢،٦) بعد تمريرها على C٤,٥:</u>

تم تطبيق المعطيات الموجودة في الجدول رقم (٢،٦) على خوارزمية Rule٣ من أجل تشذيب القوانين/M.S.Aksoy,١٩٩٥/ [٢١] وذلك من خلال أخذ القوانين الناتجة عن C٤,٥ الجدول رقم (٣،٦) وإعطائها لخوارزمية Rule٣ لتشذيبها فنتجت القوانين التالية:

جدول (١١،٦)

| Weather | Tress | Temperature | season |
|---|---|---|---|
| * | * | High | Summer |
| * | Leafless | Average | Autumn |
| * | Yellow | * | Autumn |
| * | green | average | spring |
| * | Leafless | low | Winter |
| snowy | * | * | Winter |
| rainy | Green | Low | Spring |

بمقارنة الجدول رقم (١١،٦) السابق مع الجدول رقم (٤،٦) نجد أن القوانين في الجدول (١١،٦) هي أقل بواحد وهي أكثر تشذيباً وأشمل من النتائج المأخوذة من الجدول (١١،٥).

b) <u>تطبيق Rule٣ على بيانات الجدول (٥،٦) بعد تمريرها على C٤,٥:</u>

بنتيجة تطبيق خوارزمية Rule٣ على المثال في الجدول رقم (٥،٦) من خلال أخذ القوانين الناتجة من خوارزمية C٤,٥ في الجدول رقم (٦،٦) ينتج الجدول التالي:

**جدول (١٢،٦)**

| Car-size | Fuel | Max-speed | Acceleration |
|---|---|---|---|
| * | Propane | * | Good |
| * | * | Low | Good |
| * | Petrol | * | Excellent |
| * | Diesel | High | Good |
| * | Diesel | Average | Poor |

وبمقارنة النتائج الحاصلة من الجدول رقم (١٢،٦) مع النتائج الحاصلة من التعديل المقترح في الجدول رقم (٧،٦) نجد أن التعديل المقترح أكثر تشذيباً.

c) <u>تطبيق Rule٣ على بيانات الجدول /٦، ٨/ بعد تمريرها على C٤,٥:</u>

وبنتيجة تطبيق خوارزمية Rule٣ على المثال في الجدول رقم (٨،٦) من خلال أخذ القوانين الناتجة من خوارزمية C٤,٥ في الجدول رقم (٩،٦) ينتج الجدول التالي:

**جدول (١٣،٦)**

| Age | Competition | Type | Profit |
|---|---|---|---|
| * | no | * | up |
| mid | yes | * | down |
| new | * | * | up |
| old | * | * | down |

بمقارنة النتائج من الجدول رقم (١٣،٦) مع الجدول الناتج عن التعديل في الجدول(١٠،٦) نجد أن القوانين أصبحت في التعديل أكثر تشذيباً واختصاراً من تشذيب القوانين عبر Rule٣.

### ٦-٣ مقارنة النتائج مع النتائج الناتجة من تطبيق Rule٥ على الجدول /٢،٦/:

من خلال تطبيق خوارزمية Rule٥ على بيانات الجدول /٢،٦/ تنتج القوانين كما في الجدول (١٣،٦) :

جدول (١٣،٦)

| Weather | Tress | Temperature | season |
|---|---|---|---|
| snowy | * | * | winter |
| * | yellow | * | Autumn |
| * | * | high | summer |
| * | green | average | spring |
| * | Leafless | low | Winter |
| * | leafless | average | autumn |
| rainy | green | low | spring |

وبالتالي نجد أن التعديل المقترح قد أنقص عدد القوانين أكثر من خوارزميتي Rule٣ وخوارزميـــة Rule٥.

# الفصل السابع

## ٦-١ الملخص:

مما سبق نستنتج أن خوارزمية C٤,٥ تتميز عن خوارزمية CART في أنها تسهم في إنتاج أشـــجار تملك عقداً بعدة أبناء أي أنها في كل عقدة يتم مناقشة جميع حالات وقيم خاصـــية Attribute لهـــذه العقـــدة وبالتالي هي تساهم في إنقاص عدد الحسابات لأنه لن يتم فحص خاصية Attribute التي يتم فحـــصها فـــي مستوىً لاحق كما أنها تستخدم في حساباتها لاختيار العقدة كلاً من مفهومي الأنتروبي والربح وبالتالي ستكون القوانين الناتجة قليلة وتتميز بالدقة والشمولية،وذلك على عكس خوارزمية CART التي يمكن لكل عقـــدة أن تعود وتدخل الفحص وبالتالي تصبح عقدة ليتم التفريق عندها في مستويات أدنى وذلك لأنها تولد أبناء ثنائية فقط أي أنها لا تغطي جميع أمثلتها عند هذه العقدة.

ولكن بالنسبة لخوارزمية C٤,٥ رغم دقتها ودقة تصنيفها لكنها تستهلك زمناً كبيراً لأنها تقوم بحساب الأنتروبي لكل Attribute عند كل مستوى يتم فيه اختيار العقدة الحالية لذلك تم اقتراح تعـــديل يـــساهم فـــي إنقاص عدد الحسابات في خوارزمية C٤,٥ وذلك من خلال اختيار العقدة المفحوصة للخاصية Attribute التي تحوي العدد الأكبر من القيم الفريدة وبالتالي لا يتم حساب الأنتروبي إلا للصفات التي تحوي عدداً متـــساوياً من القيم الفريدة عندها سيتم حساب الأنتروبي والربح لهم جميعهم ومما سبق نستنتج أن التعديل المقترح على خوارزمية C٤,٥ ساهم بشكل كبير وملحوظ من المنحنيات البيانية /٦-١و٦-٢/ في إنقاص عدد الحـــسابات اللازمة لحساب للأنتروبي والربح ولكنها ربما تسببت في تزايد عدد القوانين الناتجة عن بناء الشجرة.

وجدنا أيضاً رغم أن الخوارزمية C٤,٥ تتسم بدقتها ودقة تصنيفها لكن قوانينها تحتاج لتشذيب لتصبح أكثر شمولية وتغطي كافة الأمثلة لذلك تم اقتراح تعديل يساهم في تشذيب قوانين خوارزمية C٤,٥ وذلك من خلال البحث عن كافة القيم التي تحدد صنفاً واحداً من أجل كل صفة على حدة وإدخال هذه القـــيم الناتجـــة كشروط مباشرة وحذف كافة الأمثلة التي تغطيها وبعد ذلك نقوم بإدخال بقية الجدول وتمريره على خوارزمية C٤,٥ المعروفة لاستنتاج بقية القوانين ومما سبق نستنتج أن التعديل المقترح على خوارزميـــة C٤,٥ ســـاهم

بشكل كبير وملحوظ من المنحنيات البيانية /٦-٣/ /٦-٤/ /٦-٥/في إنتاج قوانين مشذبة ومقلمة وأكثر شمولية من القوانين الأصلية ومن القوانين الحاصلة من تطبيق تشذيب خوارزمية Rule٣ على القوانين الناتجة من خوارزمية C٤,٥ من خلال إنقاصه قيم الخصائص غير المناسبة بالإضافة إلى أن الزمن المستهلك لتطبيق خوارزمية C٤,٥ ومن ثم تطبيق خوارزمية Rule٣ كبير بالنسبة إلى خوارزمية C٤,٥ مع تعديلها مع العلم أن الزمن المستهلك في خوارزمية استحصال المعطيات Rule٣ كبير جداً إذا ما قورن مع الزمن الذي تستهلكه خوارزمية C٤,٥ حيث أن خوارزمية C٤,٥ تأخذ زمناً قصيراً جداً في استنباط القوانين وتعد من أسرع خوارزميات التصنيف رغم حاجتها لتشذيب قوانينها للحصول على أفضل النتائج.

يمكن تلخيص النتائج من خلال الجدولين التاليين:

**جدول (١،٧)**

| | مشكلة اللعب | | مشكلة الطقس | |
|---|---|---|---|---|
| | **C٤,٥** | **تحسين فعالية C٤,٥** | **C٤,٥** | **تحسين فعالية C٤,٥** |
| **عدد القوانين** | ٧ | ٨ | ١٠ | ١١ |
| **عدد الحسابات** | ٢٣ | ١٢ | ١٧ | ٧ |

**جدول (٢،٧)**

| | مشكلة السيارة | | مشكلة الطقس | | مشكلة stocke Market | |
|---|---|---|---|---|---|---|
| | C٤,٥ | تشذيب C٤,٥ | C٤,٥ | تشذيب C٤,٥ | C٤,٥ | تشذيب C٤,٥ |
| عدد القوانين | ٧ | ٦ | ٧ | ٥ | ٧ | ٤ |
| عدد الأمثلة الباقية | ١٢ | ٦ | ٨ | ٥ | ٩ | ٤ |

### ٦-٢ الأعمال المستقبلية:

١- تعديل خوارزمية C٤,٥ من أجل التعامل مع القيم الرقمية بشكل أكثر فعالية.

٢- تعديل خوارزمية C٤,٥ من أجل التعامل مع القيم غير المعروفة بشكل أكثر فعالية.

٣- أخذ النتائج بعد التعديلات المعمولة ثم التعامل مع خوارزميات التنقيب الأخرى المكملة للتصنيف /Clustering and Association Algorithms/ وبناء تطبيق للبحث في الويب.

# المراجع


[١] TSIEN CL, FRASER HS, LONG WJ, KENNEDY RL. "**Using classification tree and logistic regression methods to diagnose myocardial infarction**". Med info ١٩٩٨;٩:٤٩٣-٤٩٧.

[٢] TSOI A. C. PEARSON R. A., "**Comparison of three classification techniques, CART, C٤,٥ and multi-layer perceptrons**", Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, ٩٦٣ – ٩٦٩.

[٣] QUINLEN J.R, ١٩٧٩, "**Discovering rules by Induction from large collection of examples**", in D.Michie (Ed) ,EXPERT System in Micro Electronic Age,Edinburgh University Press,PP.١٦٨-٢٠١.

[٤] STEADMAN HJ, SILVER E, MONAHAN J, APELBAUM PS, ROBBINS PC, MULVEY EP, GRISSO T, ROTH LH,BANKS S. "**A classification tree approach to the development of actuarial violence risk assessment tools**". Law and Human Behavior ٢٠٠٠;٢٤:٨٣-١٠٠.

[٥]IAN H.; FRANK E.,٢٠٠٥– "**Data Mining Practical Machine Learning Tools and Techniques**". Morgan Kaufmann, ٢nd ed, San Francisco, ٥٣٠.

[٦] FRANK, E., WANG, Y., INGLIS, S., HOLMES, G. & WITTEN, I. H. ١٩٩٨, `**Using model trees for classification'**, Machine Learning ٣٢(١), ٦٣-٧٦.

[٧]UTGO, P. E., ١٩٩٧, `**Decision tree induction based on ecient tree restructuring'**, Machine Learning ٢٩, ٥.

[٨] WALLACE, C. & PATRICK, J. ١٩٩٣, `**Coding decision trees'**, Machine Learning ١١, ٧-٢٢.

QUINLAN, J. R. & RIVEST, R. L. ١٩٨٩, `Inferring decision trees using the minimum [٩] description length principle', Information and Computation ٨٠, ٢٢٧-٢٤٨.

[١٠]S. Brin and L. Page, "**The Anatomy of a Large-Scale Hyper textual Web Search Engine**," *Proc. ٧th Int'l World Wide Web Conf.* (WWW٩٨), ACM Press,New York, ١٩٩٨, pp. ١٠٧-١١٧.

[١١]J. Srivastava et al., "**Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data**,"*SIGKDD Explorations*, vol. ١, no. ٢, ٢٠٠٠, pp. ١٢-٢٣.

[١٢] S. Chakrabarti et al., "**Mining the Web's Link Structure**," *Computer,* Aug. ١٩٩٩, pp. ٦٠-٦٧

[١٣]R. Baeza-Yates and B. Ribeiro-Neto, **Modern InformationRetrieval**, Addison-Wesley, Reading, Mass.,١٩٩٩.

[١٤]S. Chakrabarti, *Mining the Web:* **"Statistical Analysis of Hypertex and Semi-Structured Data"**, Morgan Kaufmann, San Francisco, ٢٠٠٢.

[١٥]J. Han and M. Kambert, *Data Mining:"* **Concepts and Techniques**", Morgan Kaufmann, San Francisco,٢٠٠١.

[١٦]H. Yu, J. Han, and K.C-C. Chang, "**PEBL: Positive Example-Based Learning for Web Page Classification Using SVM**," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery in Databases* (KDD٠٢), ACM Press, New York, ٢٠٠٢, pp. ٢٣٩-٢٤٨.

[١٧]V. R. Borkar, K. Deshmukh, and S. Sarawagi, "**Automatic Segmentation of Text into Structured Records**," *Proc. ACM-SIGMOD Int'l Conf. Management of Data (*SIGMOD ٢٠٠١), ACM Press, New York,٢٠٠١, pp. ١٧٥-١٨٦.

[١٨]Chen, H., Shankaranararayanan, G., She, L. "**A Machine Learning Approachto Inductive Feedback, ID٣, Genetic Algorithms, Query by Examples: An Experiment Using Relevance and Simulated Annealing**." Journalfor the American Society of Information Science ٤٩,٨ (١٩٩٨): ٦٩٣-٧٠٥.

[١٩]. PHAM D.T, AKSOY M.S, ١٩٩٣-**A new algorithm for inductive learning**",Journal of System Eng.,١١٥-١٢٢.

[٢٠]. Zhang T., Ramakrishnan R. and Livny M., BIRCH: **A New Data Clustering Algorithm and Its Applications**, *International Journal of Data Mining and Knowledge Discovery*, vol. ١, No. ٢, pp ١٤١-١٨٢, ١٩٩٧.

[٢١]Aksoy M.S, **Pruning Decision Tree Using RULES-٣ Inductive Learning Algorithm**, International Journal of Mathematical and Computational Applications, vol.١٠, No. ١, pp. ١١٣-١٢٠, ٢٠٠٥.

[٢٢]Cohen, W. W, **Efficient pruning methods for separate-and-conquer rule learning systems**. *Proceeding of the ١٣th International Joint Conference on Artificial Intelligence*, Chambery, France, ٩٨٨-٩٩٤, ١٩٩٣.

[٢٣]Fürnkranz, J., **Pruning Algorithms for Rule Learning**. *Machine Learning*, Vol. ٢٧(٢),١٣٩-١٧١, ١٩٩٦.

[٢٤]IAN H.; FRANK E, **Data Mining Practical Machine Learning Tools and Techniques**. Morgan Kaufmann, ٢nd ed, San Francisco, ٥٣٠ pages , ٢٠٠٥.
[٢٥] Adamo, J., **Data Mining for Association Rules and Sequential Patterns**, Springer, New York, ٢٠٠١.

[٢٦]Jiawei Han,Kevin Chen-Chuan Chang., "**Data Mining for Web Intelligence**,"*SIGKDD Explorations*, vol. ١, no. ٢, ٢٠٠٢, pp. ٥٤-٦٠.

[٢٧]Finn, P., S. Muggleton, D. Page, A. Srinivasan, **Pharmacophore Discovery using the Inductive Logic Programming System Prolog**, Machine Learning, Special Issue on Applications and Knowledge Discovery ٣٣, no. ١ (١٩٩٨) ١٣–٤٧.

[٢٨] Thrun, S. and C. Faloutsos**, Automated Learning and Discovery**, AI Magazine (Fall ١٩٩٩) ٧٨–٨٢.

[٢٩] Dumais, S. T., J. Platt, D.Heckerman, and M. Sahami. ١٩٩٨. **Inductive learning algorithms and representations for text categorization**. In Proceedings of the ACM Seventh International Conference on Information and Knowledge Management,Bethesda, MD. New York: ACM, pp. ١٤٨–١٥٥.

## الأبحاث المنشورة

١- نجار يحيى،الحصرم حيان ، قزموز عصام ، "**تحسين فعالية شجرة التصنيف**" مجلة البحوث جامعة حلب سلسلة العلوم الهندسية، العدد ٦٢ عام ٢٠٠٨.

٢- نجار يحيى،الحصرم حيان ، قزموز عصام ، "**تشذيب قوانين شجرة التصنيف**" مجلة البحوث جامعة حلب سلسلة العلوم الهندسية، العدد ٦٣ عام ٢٠٠٨.

**الملحق A**

**وصف حساب ترتيب صفحات الويب:**

كلما كان ترتيب الصفحة أكبر تعطيها أهمية ونوعية أكبر.

ترتيب الصفحة يعرف كمايلي:

بفرض لدينا الصفحة A التي تضم مجموعة من الصفحات T١... $T_n$ التي تشير لها.

وباعتبار أن البرامتر d يمثل عامل الدمج الذي تتراوح قيمته مابين ١...٠،وغالباً ما نضع قيمته تساوي ٠,٨٥.

وبالتالي يتم حساب ترتيب الصفحة A من العلاقة التالية :

$$PR(A) = (1-d) + d(\Pr(T1)/C(T1) + ..... + PR(T_n)/C(T_n))$$

حيث أن C(A) تمثل عدد الربطات التي links التي تشير إلى خارج الصفحة.

من العلاقة السابقة نجد أن نموذج ترتيب الصفحات يعتمد على التوزيع الاحتمالي وبالتالي سيكون مجموع ترتيب كل صفحات الويب يساوي إلى الواحد.

حيث يتم حساب ترتيب الصفحة باستخدام خوارزمية تعاوديه بسيطة.

**الملحق B**

**ID٣ Algorithm**

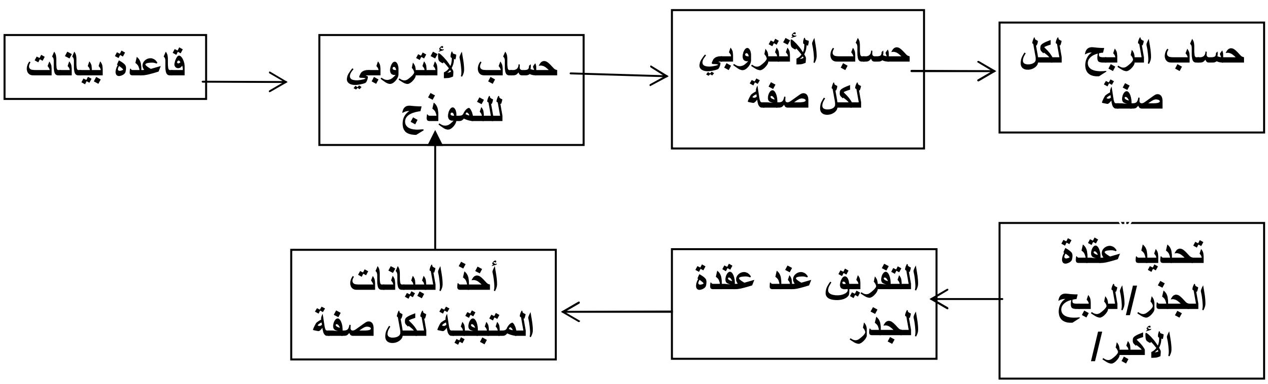
```
function ID3 (I, 0, T) {
/* I is the set of input attributes
* O is the output attribute
* T is a set of training data
*
* function ID3 returns a decision tree
*/
if (T is empty) {
return a single node with the value "Failure";
}
if (all records in T have the same value for O) {
return a single node with that value;
}
if (I is empty) {
return a single node with the value of the most frequent value of
O in T;
/* Note: some elements in this node will be incorrectly classified
*/
}
/* now handle the case where we can't return a single node */
compute the information gain for each attribute in I relative to
T;
let X be the attribute with largest Gain(X, T) of the attributes
in I;
let {x_j| j=1,2, .., m} be the values of X;
let {T_j| j=1,2, .., m} be the subsets of T when T is partitioned
according the value of X;
return a tree with the root node labelled X and
```

```
arcs labelled x_1, x_2, .., x_m, where the arcs go to the
trees ID3(I-{X}, O, T_1), ID3(I-{X}, O, T_2), .., ID3(I-{X}, O,
T_m);

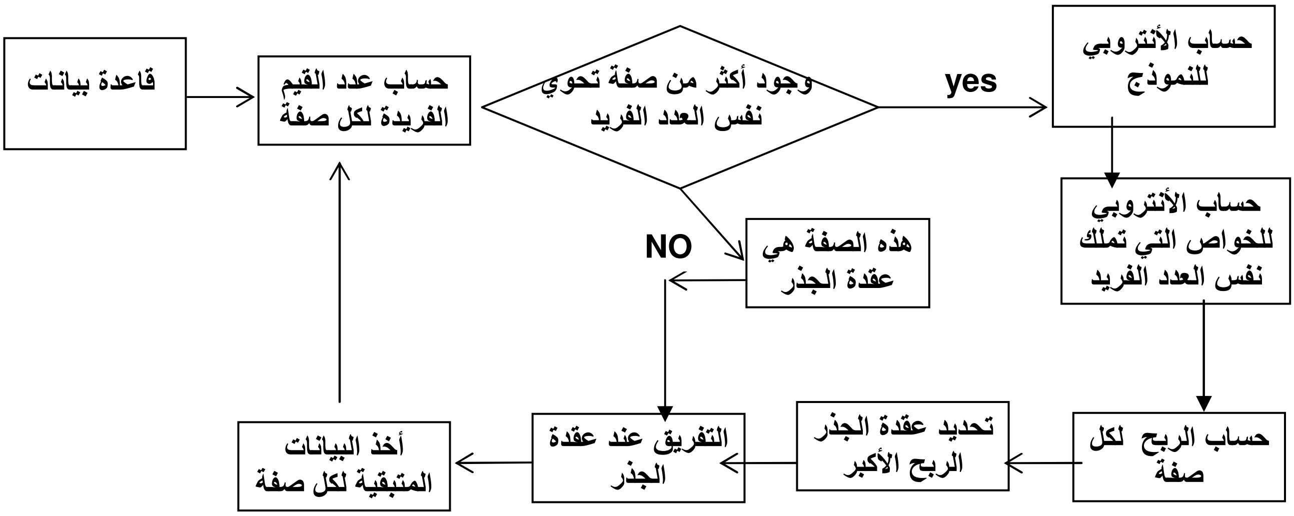
}
```

**الملحق C:**

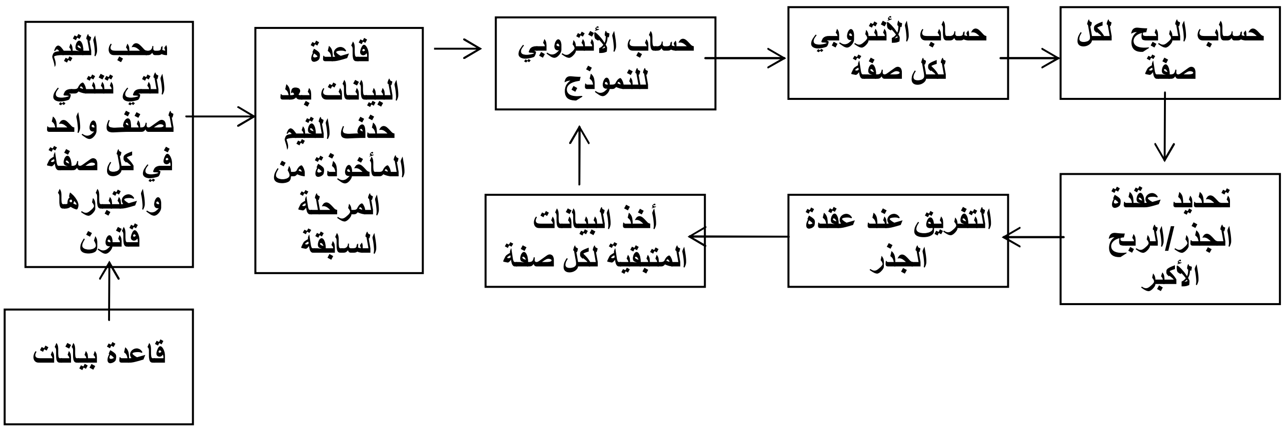
**مخططات توضح آلية عمل الخوارزمية C٤,٥ وتعديلاتها**




**مخطط مبسط لعمل خوارزمية C٤,٥**




**مخطط يوضح التعديل على خوارزمية C٤,٥ /تحسين الفعالية/**




**مخطط يوضح التعديل على خوارزمية C٤,٥ /تشذيب C٤,٥/**

## الملحق D:

**القسم العملي (برنامج بلغة C# يمثل عمل خوارزمية C٤,٥)**

```
using System;
using System.Collections;
using System.Data;

namespace ExemploID٣
{

    public class Attribute
    {
        ArrayList mValues;
        string mName;
        object mLabel;


        public Attribute(string name, string[] values)
        {
            mName = name;
            mValues = new ArrayList(values);
            mValues.Sort();
```

```
    }

    public Attribute(object Label)
    {
        mLabel = Label;
        mName = string.Empty;
        mValues = null;
    }


    public string AttributeName
    {
        get
        {
            return mName;
        }
    }

    public string[] values
    {
        get
        {
            if (mValues != null)
                return (string[])mValues.ToArray(typeof(string));
            else
                return null;
        }
    }


    public bool isValidValue(string value)
    {
        return indexValue(value) >= ٠;
    }

    public int indexValue(string value)
    {
        if (mValues != null)
            return mValues.BinarySearch(value);
        else
            return -١;
    }


    public override string ToString()
    {
        if (mName != string.Empty)
        {
            return mName;
        }
        else
        {
            return mLabel.ToString();
        }
    }
}

public class TreeNode
{
    private ArrayList mChilds = null;
    private Attribute mAttribute;

    public TreeNode(Attribute attribute)
    {
```

```
            if (attribute.values != null)
            {
                //try
                {
                    mChilds = new ArrayList(attribute.values.Length);
                    for (int i = ٠; i < attribute.values.Length; i++)
                        mChilds.Add(null);
                }//catch(Exception e)
            }
            else
            {
                mChilds = new ArrayList(١);
                mChilds.Add(null);
            }
            mAttribute = attribute;
        }


        public void AddTreeNode(TreeNode treeNode, string ValueName)
        {
            int index = mAttribute.indexValue(ValueName);
            mChilds[index] = treeNode;
        }

        public int totalChilds
        {
            get
            {
                return mChilds.Count;
            }
        }

        public TreeNode getChild(int index)
        {
            return (TreeNode)mChilds[index];
        }


        public Attribute attribute
        {
            get
            {
                return mAttribute;
            }
        }


        public TreeNode getChildByBranchName(string branchName)
        {
            int index = mAttribute.indexValue(branchName);
            return (TreeNode)mChilds[index];
        }
    }

    public class DecisionTreeID٣
    {
        private DataTable mSamples;
        private int mTotalPositivessum = ٠;
        private int mTotalPositivessp = ٠;
        private int mTotalPositivesaut = ٠;
        private int mTotal = ٠;
        private string mTargetAttribute = "result";
        private double mEntropySet = ٠,٠;
```

```
        private int countTotalPositivesaut(DataTable samples)
        {
            int result = ٠;

            foreach (DataRow aRow in samples.Rows)
            {
                if ((string)aRow[mTargetAttribute] == "poor")
                    result++;
            }

            return result;
        }
        private int countTotalPositivessp(DataTable samples)
        {
            int result = ٠;

            foreach (DataRow aRow in samples.Rows)
            {
                if ((string)aRow[mTargetAttribute] == "excellent")
                    result++;
            }

            return result;
        }
        private int countTotalPositivessum(DataTable samples)
        {
            int result = ٠;

            foreach (DataRow aRow in samples.Rows)
            {
                if ((string)aRow[mTargetAttribute] == "good")
                    result++;
            }

            return result;
        }

        private double calcEntropy(int p١, int p٢, int p٣)
        {
            int total = p١ + p٢ + p٣ ;
            double ratioP١ = (double)p١ / total;
            double ratiop٢ = (double)p٢ / total;
            double ratioP٣ = (double)p٣ / total;
            if (ratioP١ != ٠)
                ratioP١ = -(ratioP١) * System.Math.Log(ratioP١, ٢);
            if (ratiop٢ != ٠)
                ratiop٢ = -(ratiop٢) * System.Math.Log(ratiop٢, ٢);
            if (ratioP٣ != ٠)
                ratioP٣ = -(ratioP٣) * System.Math.Log(ratioP٣, ٢);
            double result = ratioP١ + ratiop٢ + ratioP٣ ;

            return result;
        }


        private void getValuesToAttribute(DataTable samples, Attribute
attribute, string value, out int p١, out int p٢, out int p٣)
        {
            p١ = ٠;
            p٢ = ٠;
            p٣ = ٠;

            foreach (DataRow aRow in samples.Rows)
```

```
            {
                if (((string)aRow[attribute.AttributeName] == value))
                    if ((string)aRow[mTargetAttribute] == "good")
                        p١++;
                    else if ((string)aRow[mTargetAttribute] == "excellent")
                        p٢++;
                    else if ((string)aRow[mTargetAttribute] == "poor")
                        p٣++;

            }
        }

        private double gain(DataTable samples, Attribute attribute)
        {
            string[] values = attribute.values;
            double sum = ٠,٠;

            for (int i = ٠; i < values.Length; i++)
            {
                int p١, p٢, p٣;

                p١ = p٢ = p٣ = ٠;

                getValuesToAttribute(samples, attribute, values[i], out p١, out
p٢, out p٣);

                double entropy = calcEntropy(p١, p٢, p٣);
                sum += -(double)(p١ + p٢ + p٣ ) / mTotal * entropy;
            }
            return mEntropySet + sum;
        }

        private Attribute getBestAttribute(DataTable samples, Attribute[]
attributes)
        {
            double maxGain = ٠,٠;
            Attribute result = null;

            foreach (Attribute attribute in attributes)
            {
                double aux = gain(samples, attribute);
                if (aux > maxGain)
                {
                    maxGain = aux;
                    result = attribute;
                }
            }
            return result;
        }


        private bool allSamplesPositivesaut(DataTable samples, string
targetAttribute)
        {
            foreach (DataRow row in samples.Rows)
            {
                if ((string)row[targetAttribute] == "good" ||
(string)row[targetAttribute] == "excellent")
                    return false;
            }

            return true;
        }
```

```
        private bool allSamplesPositivessum(DataTable samples, string
targetAttribute)
        {
            foreach (DataRow row in samples.Rows)
            {
                if ((string)row[targetAttribute] == "poor" ||
(string)row[targetAttribute] == "excellent")
                    return false;
            }

            return true;
        }
        private bool allSamplesPositivessp(DataTable samples, string
targetAttribute)
        {
            foreach (DataRow row in samples.Rows)
            {
                if ((string)row[targetAttribute] == "good" ||
(string)row[targetAttribute] == "poor")
                    return false;
            }

            return true;
        }

        private ArrayList getDistinctValues(DataTable samples, string
targetAttribute)
        {
            ArrayList distinctValues = new ArrayList(samples.Rows.Count);

            foreach (DataRow row in samples.Rows)
            {
                if (distinctValues.IndexOf(row[targetAttribute]) == -١)
                    distinctValues.Add(row[targetAttribute]);
            }

            return distinctValues;
        }


        private object getMostCommonValue(DataTable samples, string
targetAttribute)
        {
            ArrayList distinctValues = getDistinctValues(samples,
targetAttribute);
            int[] count = new int[distinctValues.Count];

            foreach (DataRow row in samples.Rows)
            {
                int index = distinctValues.IndexOf(row[targetAttribute]);
                count[index]++;
            }

            int MaxIndex = ٠;
            int MaxCount = ٠;

            for (int i = ٠; i < count.Length; i++)
            {
                if (count[i] > MaxCount)
                {
                    MaxCount = count[i];
                    MaxIndex = i;
                }
```

```
            }

            return distinctValues[MaxIndex];
        }


        private TreeNode internalMountTree(DataTable samples, string
targetAttribute, Attribute[] attributes)
        {
            if (allSamplesPositivessum(samples, targetAttribute) == true)
                return new TreeNode(new Attribute("good"));

            if (allSamplesPositivessp(samples, targetAttribute) == true)
                return new TreeNode(new Attribute("excellent"));
            if (allSamplesPositivesaut(samples, targetAttribute) == true)
                return new TreeNode(new Attribute("poor"));

            if (attributes.Length == ٠)
                return new TreeNode(new Attribute(getMostCommonValue(samples,
targetAttribute)));

            mTotal = samples.Rows.Count;
            mTargetAttribute = targetAttribute;
            mTotalPositivesaut = countTotalPositivesaut(samples);
            mTotalPositivessum = countTotalPositivessum(samples);
            mTotalPositivessp = countTotalPositivessp(samples);
          //  mTotalPositiveswin = countTotalPositiveswin(samples);
            mEntropySet = calcEntropy(mTotalPositivesaut, mTotalPositivessum,
mTotalPositivessp);

            Attribute bestAttribute = getBestAttribute(samples, attributes);

            TreeNode root = new TreeNode(bestAttribute);

            DataTable aSample = samples.Clone();

            foreach (string value in bestAttribute.values)
            {

                aSample.Rows.Clear();

                DataRow[] rows = samples.Select(bestAttribute.AttributeName + "
= " + "'" + value + "'");

                foreach (DataRow row in rows)
                {
                    aSample.Rows.Add(row.ItemArray);
                }

                ArrayList aAttributes = new ArrayList(attributes.Length - ١);
                for (int i = ٠; i < attributes.Length; i++)
                {
                    if (attributes[i].AttributeName !=
bestAttribute.AttributeName)
                        aAttributes.Add(attributes[i]);
                }


                if (aSample.Rows.Count == ٠)
                {
                    return new TreeNode(new
Attribute(getMostCommonValue(aSample, targetAttribute)));
                }
                else
```

```
                {
                    DecisionTreeID٣ dc٣ = new DecisionTreeID٣();
                    TreeNode ChildNode = dc٣.mountTree(aSample, targetAttribute, 
(Attribute[])aAttributes.ToArray(typeof(Attribute)));
                    root.AddTreeNode(ChildNode, value);
                }
            }

            return root;
        }



        public TreeNode mountTree(DataTable samples, string targetAttribute, 
Attribute[] attributes)
        {
            mSamples = samples;
            return internalMountTree(mSamples, targetAttribute, attributes);
        }
    }

    class ID٣Sample
    {

        public static void printNode(TreeNode root, string tabs)
        {
            Console.WriteLine(tabs + '|' + root.attribute + '|');

            if (root.attribute.values != null)
            {
                for (int i = ٠; i < root.attribute.values.Length; i++)
                {
                    Console.WriteLine(tabs + "\t" + "<" + 
root.attribute.values[i] + ">");
                    TreeNode childNode = 
root.getChildByBranchName(root.attribute.values[i]);
                    printNode(childNode, "\t" + tabs);
                }
            }
        }


        static DataTable getDataTable()
        {
            DataTable result = new DataTable("samples");
            DataColumn column = result.Columns.Add("carsize");
            column.DataType = typeof(string);

            column = result.Columns.Add("fuel");
            column.DataType = typeof(string);

            column = result.Columns.Add("maxspeed");
            column.DataType = typeof(string);


            column = result.Columns.Add("result");
            column.DataType = typeof(string);

            result.Rows.Add(new object[] { "large", "diesel", "high", "good" });
           result.Rows.Add(new object[] { "large", "propane", "high", "good" });
            result.Rows.Add(new object[] { "compact", "petrol", "high", 
"excellent" });
            result.Rows.Add(new object[] { "large", "petrol", "high", 
"excellent" });
```

```
            result.Rows.Add(new object[] { "medium", "diesel", "low", "good" });
             result.Rows.Add(new object[] { "compact", "petrol", "low", "good" });
            result.Rows.Add(new object[] { "medium", "petrol", "average",
"excellent" });
            result.Rows.Add(new object[] { "medium", "diesel", "average", "poor"
});
            return result;

        }

        [STAThread]
        static void Main(string[] args)
        {
            Attribute carsize = new Attribute("carsize", new string[] { "large",
"compact", "medium" });
            Attribute fuel = new Attribute("fuel", new string[] { "diesel",
"petrol" ,"propane"});
            Attribute maxspeed = new Attribute("maxspeed", new string[] {
"average", "high","low" });


            Attribute[] attributes = new Attribute[] { carsize, fuel, maxspeed
};

            DataTable samples = getDataTable();

            DecisionTreeID٣ id٣ = new DecisionTreeID٣();
            TreeNode root = id٣.mountTree(samples, "result", attributes);

            printNode(root, "");

            Console.Read();
        }
    }
```


**Abstract**:

CART algorithm's make decision tree with tow branches at each node but the attribute that entered at any node may be enter in another node in lower level at the tree that means this attribute will not quit from the solution and may be appear again to test all the remaining values.

Wile the C٤,٥ algorithm can makes the multi branches trees that it means the attribute which will be the node it will test all the value in the attribute then it will quit from the solution in the lower level and it will deleted and did not appear again.

I suggested changes at C٤,٥ algorithm to make the entropy and gain calculation become smaller.

In this changes I suggested to choose the node from the attributes that have the more unique value in it.

But if we have more one attributes have the same number of the unique values we must calculate the entropy for this attributes.

This changes may reduce the number of calculation of the entropy and reduce the consumer time for calculating but the result rules still cover all the examples but it will make the rules changes.

But the rules that makes by C٤,٥ algorithm need for pruning to be more general as it contains no irrelevant attribute-value.

I suggested changes at C٤,٥ algorithm for pruning the decision trees rules. and this changes makes the algorithm search at each attribute about the value that gives only one class then add this value into the result rules after then delete all the examples that gives this state from the table then take the remained data to C٤,٥ algorithm to gives the another rules.

As we saw from the article This changes make the rules more general as contains no irrelevant attribute-value.

Then make a comparison between the result rules that obtained by applied the output of C٤,٥ rules to Rule٣ algorithm with the result that obtained from applied the change on the C٤,٥ algorithm.